

Chapter 4:

Software Architecture

Design

4 Software Architecture Design

4.1 Introduction

This document describes the technical and UI design of the Beauty Plus system. It includes the architectural design and the detailed design. It also includes the design of operational database and data warehouse.

- The architectural design describes the overall architecture of the system, and the architecture of each main component and subsystem.
- The detailed design describes static and dynamic structure for each component and function.
- It includes class diagrams, class explanations, and sequence diagrams of the main use cases.
- The database design describes the relationship between entities, and details of each entity.

4.2 System Overview

4.2.1 Choosing system architecture design:

As the overall architecture of Beauty Plus System, the architecture we supposed that suitable for the situation is Multiple Tier- Architecture.

Beauty plus System has the application would be deployed on Android OS devices, the application is the presentation tier . Client send request to application server, application server then handle request by itself or by communicating with search engine or data warehouse server and send appropriate response to client.

Each server will handle their own data source of the data tier, which consists of operational database (for application server), data warehouse database (for data warehouse server) and documents (for search engine).

Finally, Paypal and Google mail services, which are used by our system, are placed on the external service component, independent from our three-tier system.

The overview of the system architecture is as following:

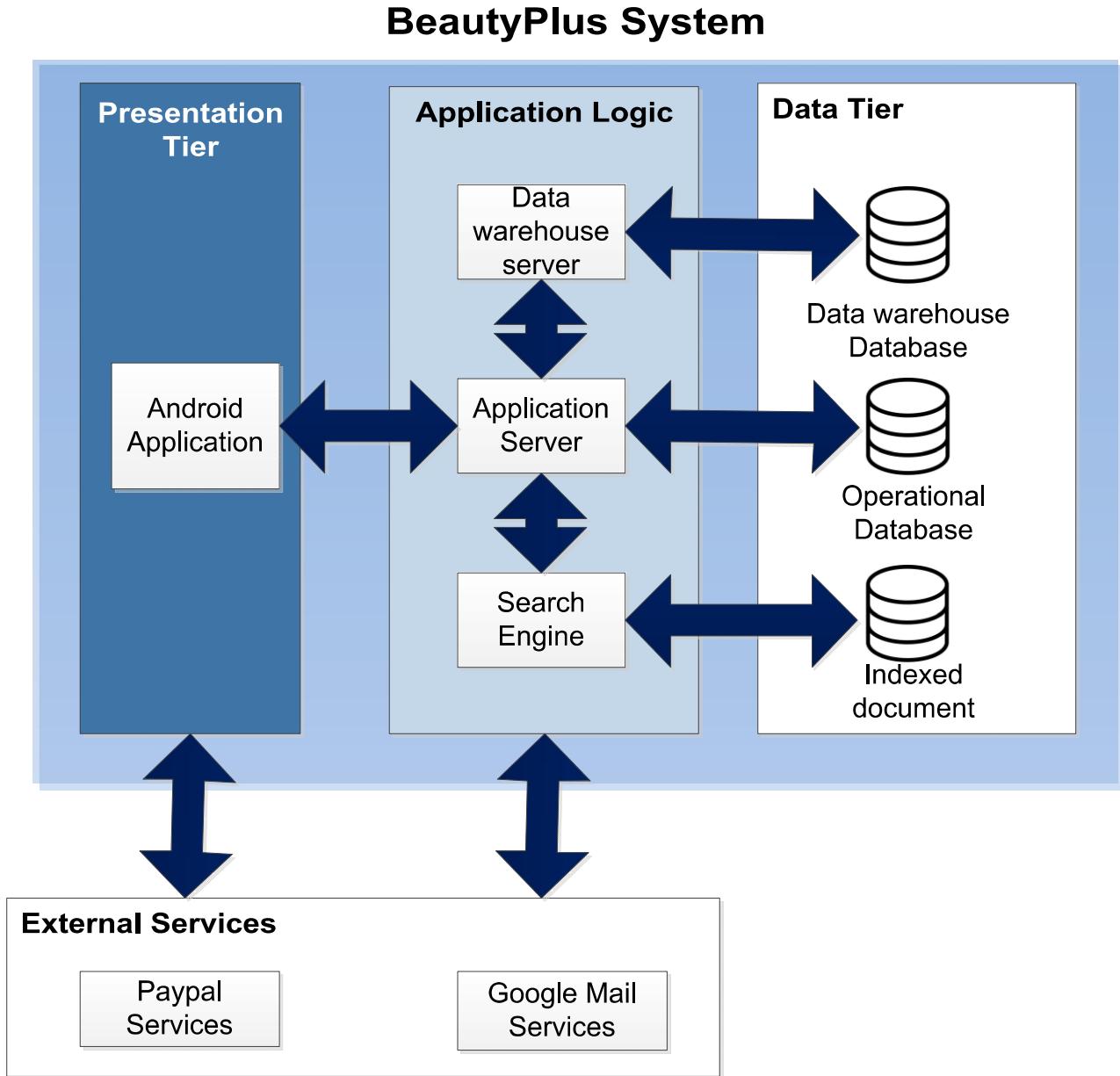


Figure 4.1: Multiple tiers architecture model of the BeautyPlus system

4.2.2 Choosing the client architecture design:

After analyzing SRS document and studying the Android platform, we had decided to choose the MVC Architecture for developing Beauty Plus application on Android OS Smart phone.

4.2.3 Choosing the server architecture design:

After analyzing SRS document and studying the Java Web Platform, we had decided to choose the Three-tier architecture for developing our application server.

4.3 System Architecture and Architecture Design

4.3.1 Logical view:

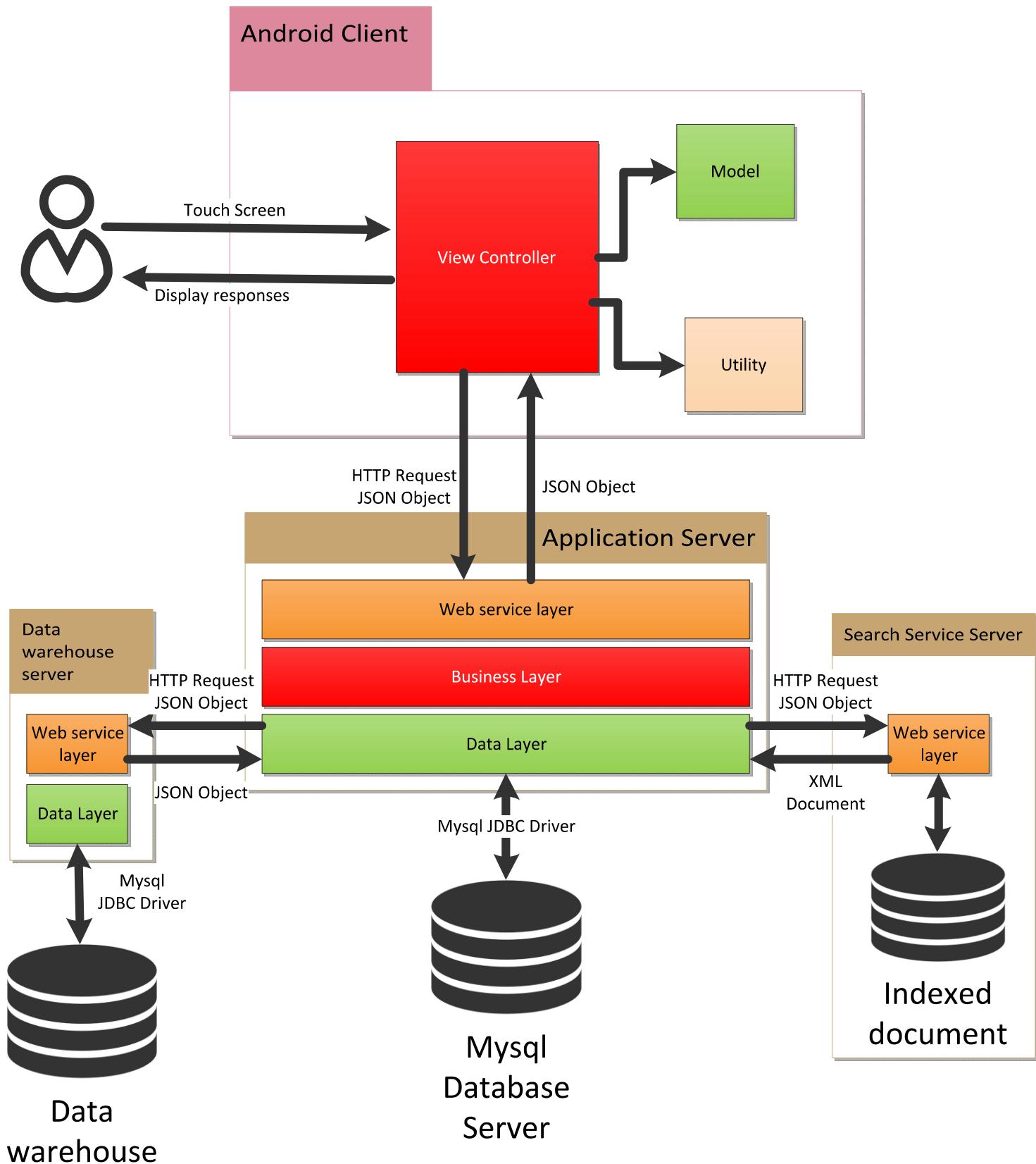


Figure 4.2 System logical design

4.3.2 Software Architecture:

4.3.2.1 Client architecture:

4.3.2.1.1 Client architecture overview:

4.3.2.1.1.1 View package:

- Present the Model to the users in an appropriate interface.
- Receive user's gesture and allows user to manipulate data.
- Does not store any data except to cache state.
- Easily reusable and configurable to display different data.

4.3.2.1.1.2 Model package:

- Manages the application data and state.
- Not concerned with user interface or representation.
- Same model will be reusable, unchanged in different interface.

4.3.2.1.1.3 Controller package:

- Intermediary between Model and View.
- Update the view when the model changes.
- Call the model when user manipulates with view.
- Typically where the application logic lives.

4.3.2.2 Application server architecture:

4.3.2.2.1 Application server overview:

4.3.2.2.1.1 Top tier:

In our system, the application server has 3 main jobs:

- Retrieve data from database as the client request
- Update database as the client request
- Update search engines and data warehouse data to match with the operational database.

After our research, we have decided that the most suitable approach for our application server to do those jobs is as a JAVA RESTful Web Service.

As we don't develop a true data-representation component on the server, the web services will be on top of our three-tier. It will act as a communicator between the business process at server and the presentation process at client.

On the other hand, we need a tool to control our server as well as our system operation, which is often called an administration tool. Due to lack of time and resources, we will develop it as a simple web site with minimal function. It need to be placed on the top layer

too because it acts as an communicator through which an administrator can get access to the business process.

4.3.2.2.1.2 Business layer:

This layer consists of business handler classes, each of which contains needed business functions. When needed, the top tier can instantiate corresponding class and use its function.

4.3.2.2.1.3 Data layer:

This layer is in charge of communicating with other component than the client. Those components are:

- Database server: this communication is done by MySQL JDBC driver
- Solr Search Services: this communication is done by http POST and GET request
- Data warehouse server: this communication is done by http POST and GET request

4.3.2.2.2 Application server basic structure:

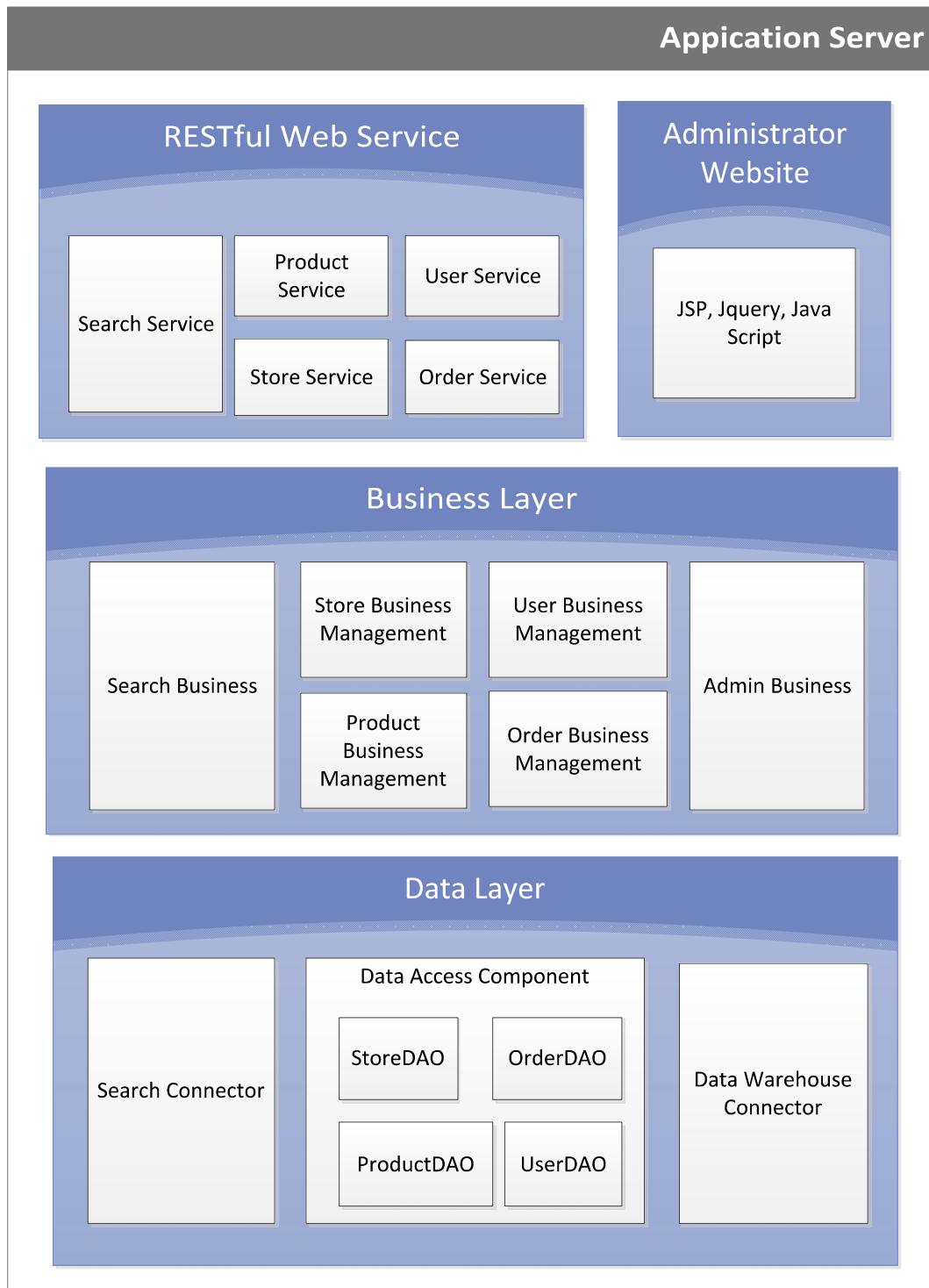


Figure 4.3 Application Server Structure

4.3.2.3 Search Services:

Searching plays an important part in our system operation but it can take up many resources from the server. In order to release the stress for our operational database as well as our application server, we have decided to implement a stand-alone search engine on a separated server.

After some researches, we have found a suitable solution for the search engines, which is Apache Solr.

4.3.3 Communication Architecture:

4.3.3.1 Communication between client and application server:

In this communication, the client will send a Http Request to server with or without data in JSON format and then receive a response without or with data also in JSON format.

4.3.3.1.1 Get data from server using http GET request:

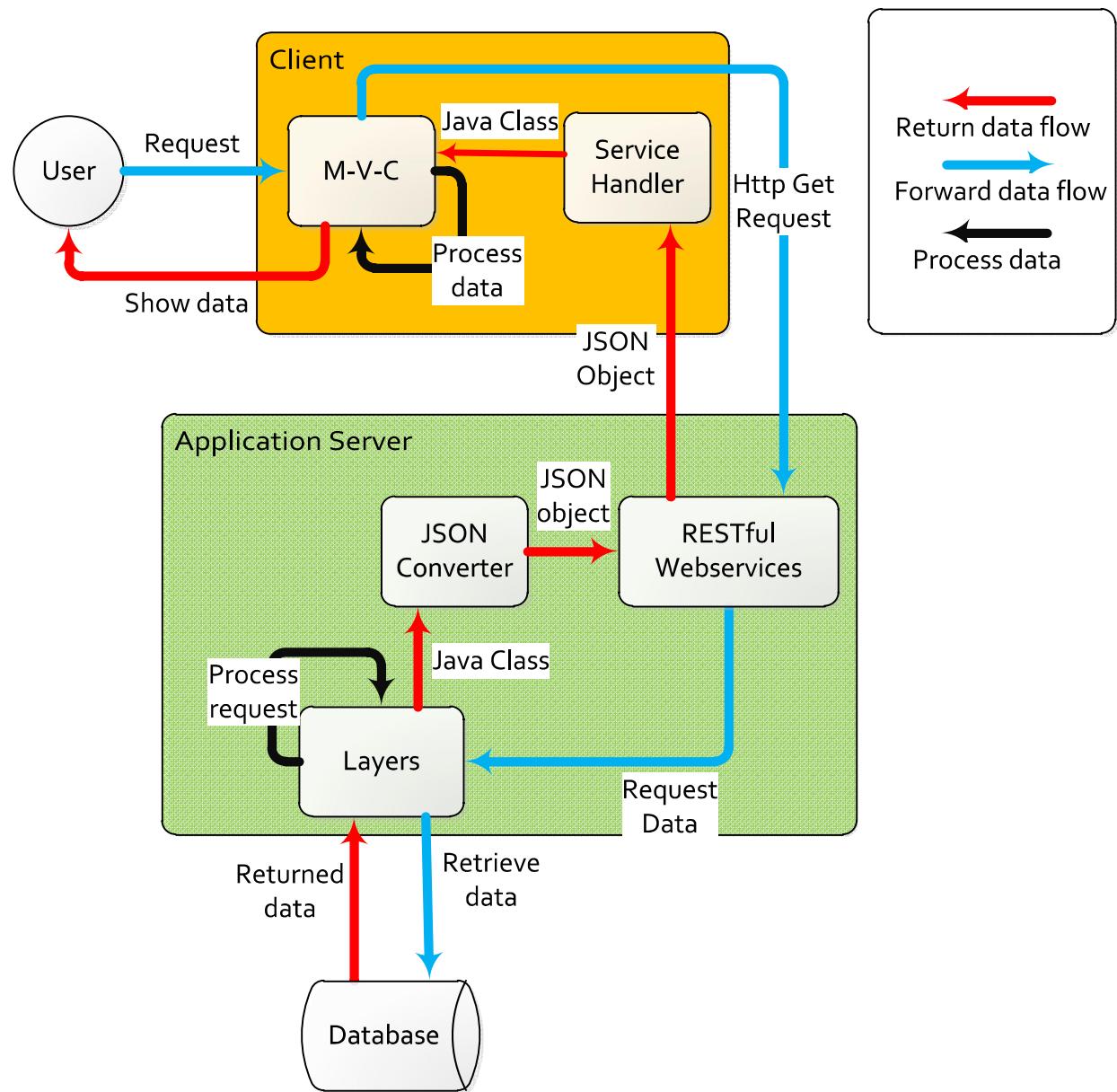


Figure 4.4 Client-Server communication diagram for GET request

4.3.3.1.2 Update/Insert data into database using http POST request:

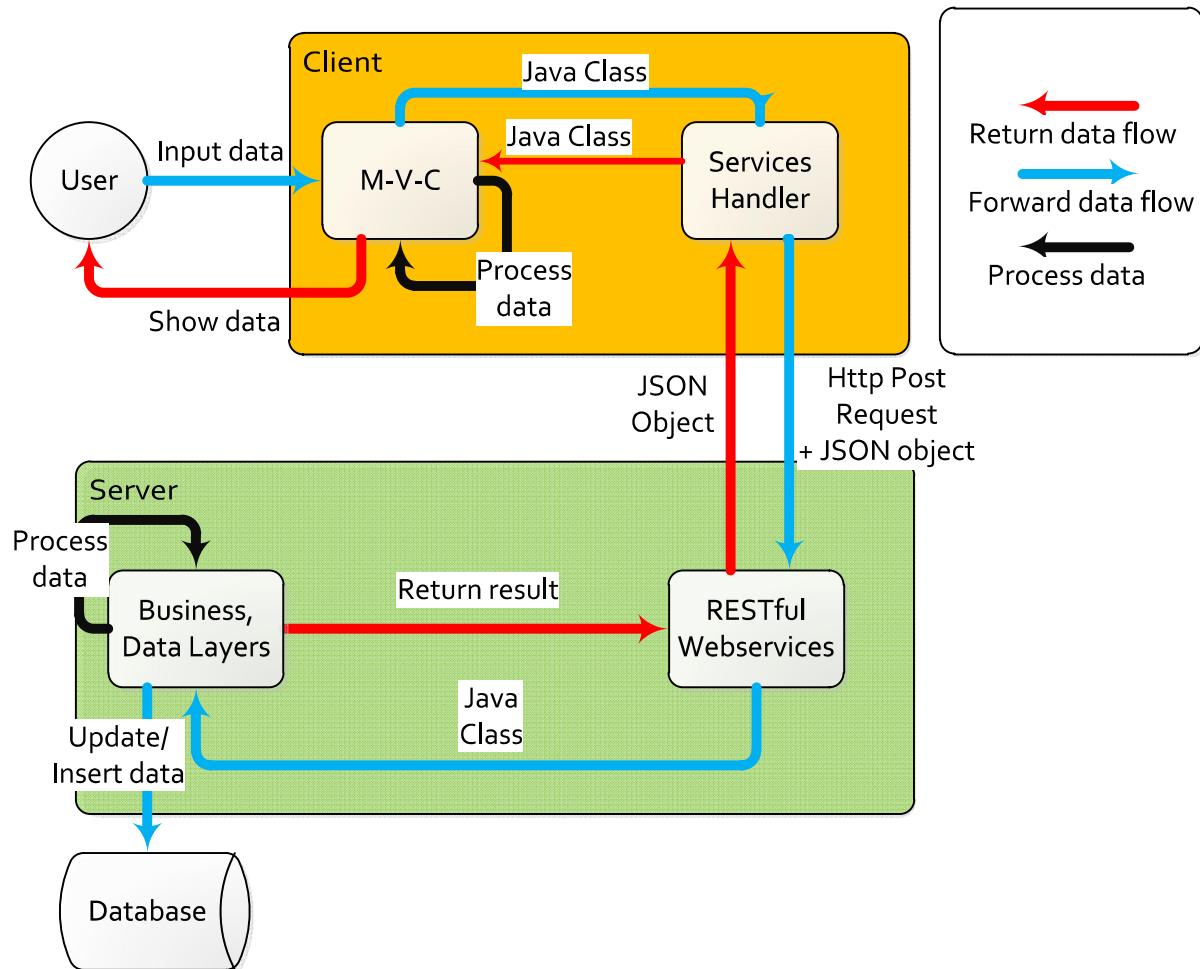


Figure 4.5 Client-Server communication diagram for POST request

4.3.3.2 Communication with search server to search for a product:

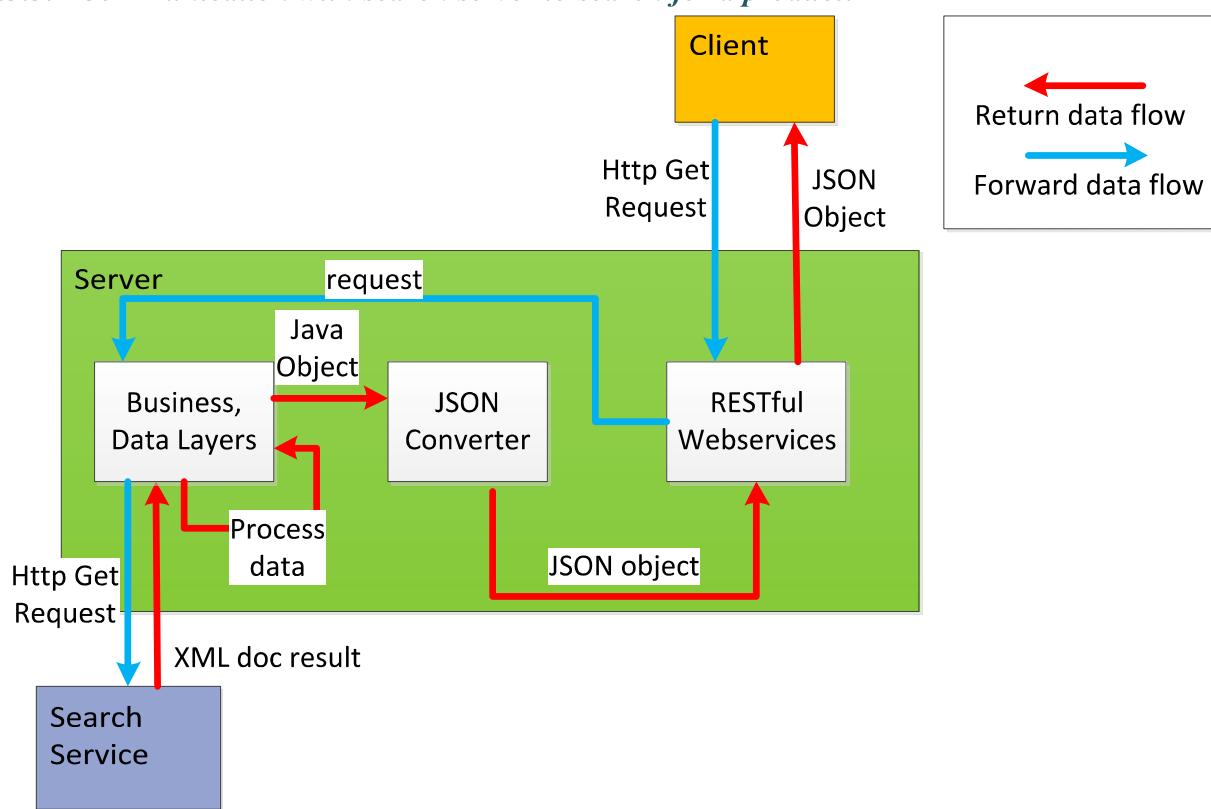


Figure 4.6: Application Server – Search server communication diagram

4.3.4 Data Architecture:

4.3.4.1 Entity Relationship Classdiagram:

Software Architecture Design

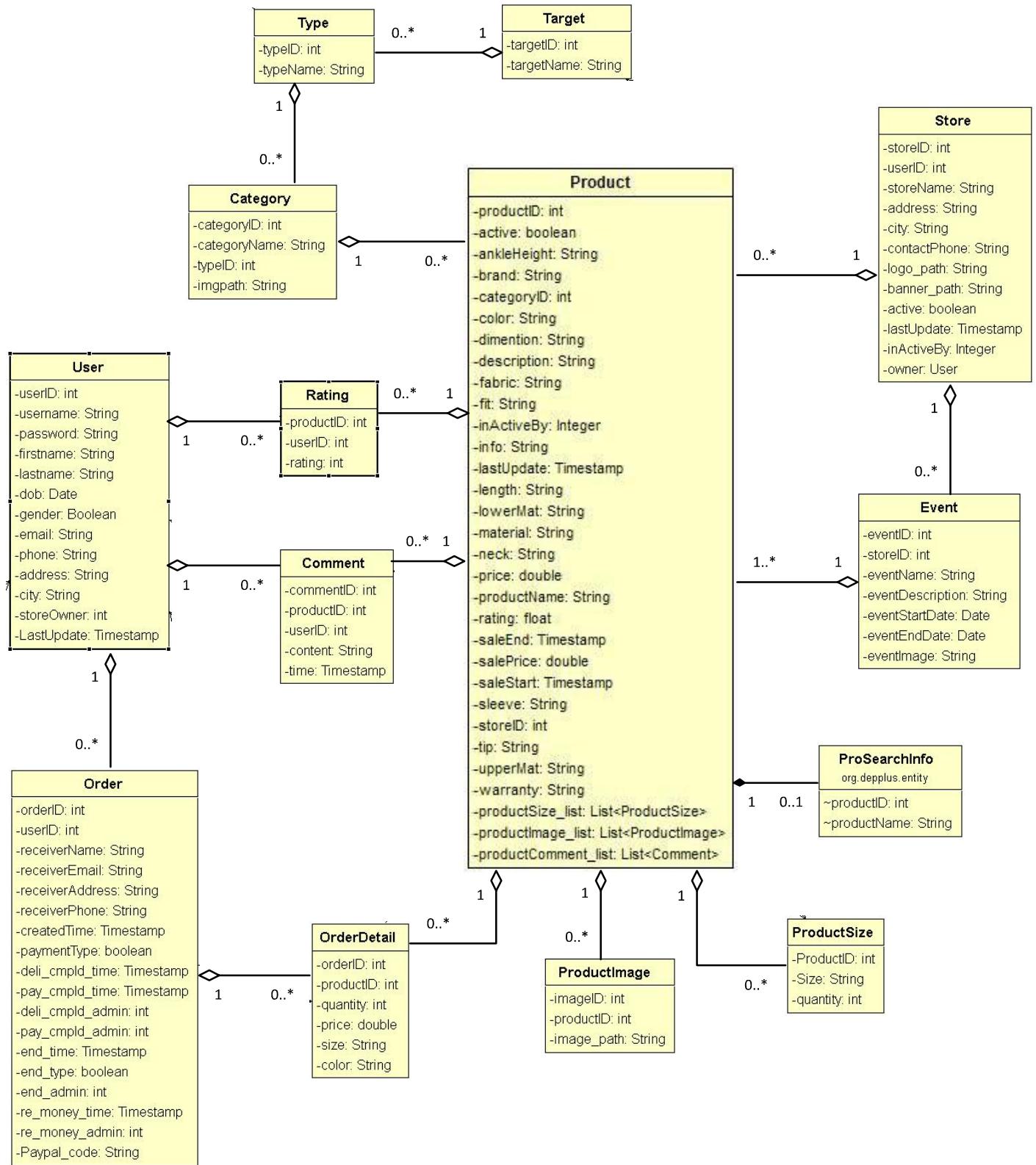


Figure 4.7: Entity Relationship class diagram

4.4 System Design:

4.4.1 Database Design:

4.4.1.1 Operational database design:

4.4.1.1.1 Relation Diagram:

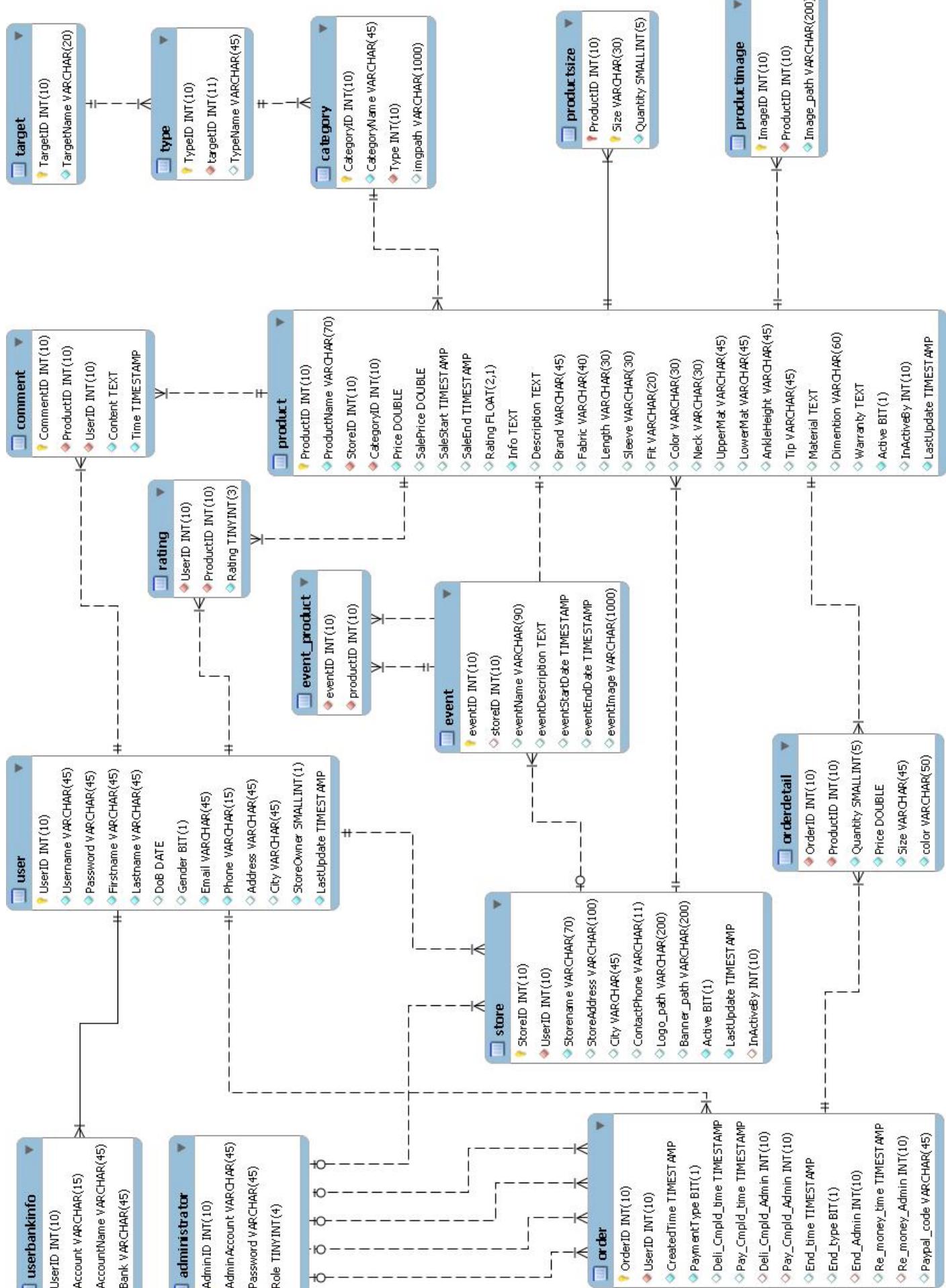


Figure 4.8 Database design

4.4.1.1.2 Table detailed design:

4.4.1.1.2.1 Note:

Bold Italiccolumn indicate **primary key** column.

Bold column indicate **unique** column.

4.4.1.1.2.2 Table administrator

Column	Type	Is Nullable?	Default
AdminID	int(10)	No	
AdminAccount	varchar(45)	No	
Password	varchar(45)	No	
Role	tinyint(4)	No	

4.4.1.1.2.3 Table category

Column	Type	Is Nullable?	Default
CategoryID	int(10)	No	
CategoryName	varchar(45)	No	
Type	int(10)	No	
Imgpath	varchar(1000)	Yes	NULL

4.4.1.1.2.4 Table comment

Column	Type	Is Nullable?	Default
CommentID	int(10)	No	
ProductID	int(10)	No	
UserID	int(10)	No	
Content	text	No	
Time	timestamp	No	CURRENT_TIMESTAMP

4.4.1.1.2.5 Table event

Column	Type	Is Nullable?	Default
eventID	int(10)	No	
storeID	int(10)	Yes	NULL
eventName	varchar(90)	Yes	NULL
eventDescription	text	Yes	NULL
eventStartDate	timestamp	Yes	NULL
eventEndDate	timestamp	Yes	NULL

eventImage	varchar(1000)	Yes	NULL
------------	---------------	-----	------

4.4.1.1.2.6 Table *event_product*

Column	Type	Is Nullable?	Default
eventID	int(10)	No	
productID	int(10)	No	

4.4.1.1.2.7 Table *order*

Column	Type	Is Nullable?	Default
<i>OrderID</i>	int(10)	No	
UserID	int(10)	No	
ReceiverName	varchar(90)	Yes	NULL
ReceiverEmail	varchar(90)	Yes	NULL
ReceiverAddress	varchar(150)	Yes	NULL
ReceiverPhone	varchar(12)	Yes	NULL
CreatedTime	timestamp	No	CURRENT_TIMESTAMP
PaymentType	bit(1)	No	
Deli_Cmpld_time	timestamp	Yes	NULL
Pay_Cmpld_time	timestamp	Yes	NULL
Deli_Cmpld_Admin	int(10)	Yes	NULL
Pay_Cmpld_Admin	int(10)	Yes	NULL
End_time	timestamp	Yes	NULL
End_type	bit(1)	Yes	NULL
End_Admin	int(10)	Yes	NULL
Re_money_time	timestamp	Yes	NULL
Re_money_Admin	int(10)	Yes	NULL
Paypal_code	varchar(45)	Yes	NULL

4.4.1.1.2.8 Table *orderdetail*

Column	Type	Is Nullable?	Default
OrderID	int(10)	No	
ProductID	int(10)	No	
Quantity	smallint(5)	No	
Price	Double	No	
Size	varchar(45)	No	

color	varchar(50)	No	
Time	Timestamp	No	CURRENT_TIMESTAMP
Content	varchar(100)	No	

4.4.1.1.2.9 Table product

Column	Type	Is Nullable?	Default
ProductID	int(10)	No	
ProductName	varchar(70)	No	
StoreID	int(10)	No	
CategoryID	int(10)	No	
Price	Double	No	0
SalePrice	Double	Yes	NULL
SaleStart	timestamp	Yes	NULL
SaleEnd	timestamp	Yes	NULL
Rating	float(2,1)	Yes	NULL
Info	Text	No	
Description	Text	Yes	NULL
Brand	varchar(45)	Yes	NULL
Fabric	varchar(40)	Yes	NULL
Length	varchar(30)	Yes	NULL
Sleeve	varchar(30)	Yes	NULL
Fit	varchar(20)	Yes	NULL
Color	varchar(30)	Yes	NULL
Neck	varchar(30)	Yes	NULL
UpperMat	varchar(45)	Yes	NULL
LowerMat	varchar(45)	Yes	NULL
AnkleHeight	varchar(45)	Yes	NULL
Tip	varchar(45)	Yes	NULL
Material	Text	Yes	NULL
Dimention	varchar(60)	Yes	NULL
Warranty	Text	Yes	NULL
Active	bit(1)	No	b'1'
InActiveBy	int(10)	Yes	NULL
LastUpdate	timestamp	No	CURRENT_TIMESTAMP

4.4.1.1.2.10 Table productimage

Column	Type	Is Nullable?	Default
ImageID	int(10)	No	
ProductID	int(10)	No	
Image_path	varchar(200)	No	

4.4.1.1.2.11 Table productsize

Column	Type	Is Nullable?	Default
ProductID	int(10)	No	
Size	varchar(30)	No	
Quantity	smallint(5)	No	0

4.4.1.1.2.12 Table rating

Column	Type	Is Nullable?	Default
UserID	int(10)	No	
ProductID	int(10)	No	
Rating	tinyint(3)	No	

4.4.1.1.2.13 Table store

Column	Type	Is Nullable?	Default
StoreID	int(10)	No	
UserID	int(10)	No	
Storename	varchar(70)	No	
StoreAddress	varchar(100)	Yes	NULL
City	varchar(45)	Yes	NULL
ContactPhone	varchar(11)	Yes	NULL
Logo_path	varchar(200)	Yes	NULL
Banner_path	varchar(200)	Yes	NULL
Active	bit(1)	No	b'1'
LastUpdate	timestamp	No	CURRENT_TIMESTAMP
InActiveBy	int(10)	Yes	NULL

4.4.1.1.2.14 Table target

Column	Type	Is Nullable?	Default
TargetID	int(10)	No	
TargetName	varchar(20)	No	

4.4.1.1.2.15 Table type

Column	Type	Is Nullable?	Default
TypeID	int(10)	No	
targetID	int(11)	No	
TypeName	varchar(45)	Yes	NULL

4.4.1.1.2.16 Table user

Column	Type	Is Nullable?	Default
UserID	int(10)	No	
Username	varchar(45)	No	
Password	varchar(45)	No	
Firstname	varchar(45)	No	
Lastname	varchar(45)	No	
DoB	Date	Yes	NULL
Gender	bit(1)	Yes	NULL
Email	varchar(45)	No	
Phone	varchar(15)	No	
Address	varchar(45)	Yes	NULL
City	varchar(45)	Yes	NULL
StoreOwner	smallint(1)	No	0
LastUpdate	timestamp	No	CURRENT_TIMESTAMP

4.4.1.1.2.17 Table userbankinfo

Column	Type	Is Nullable?	Default
UserID	int(10)	No	
Account	varchar(15)	No	
AccountName	varchar(45)	No	
Bank	varchar(45)	Yes	NULL

4.4.2 Data warehouse database design:

4.4.2.1 Relation diagram:

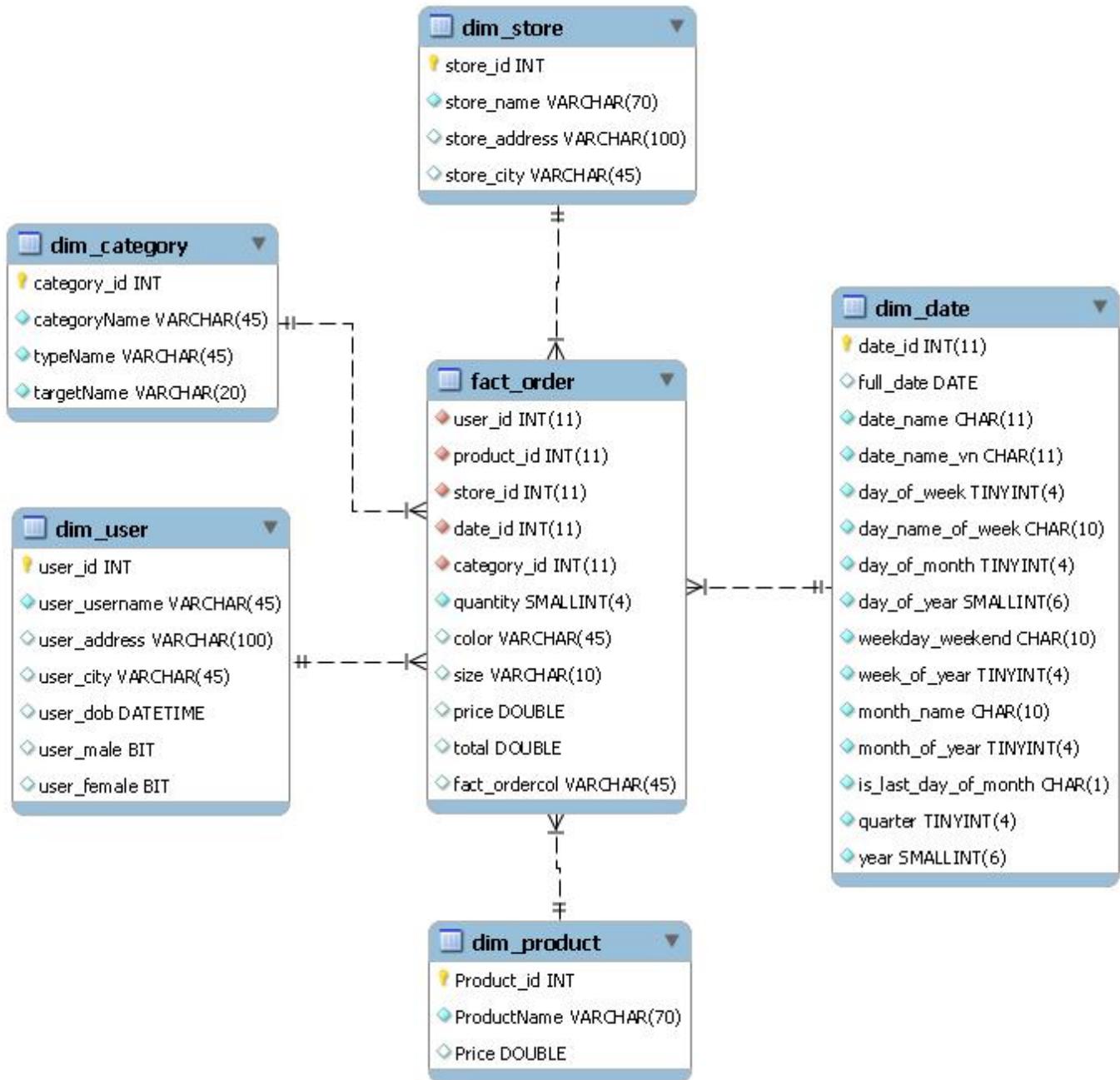


Figure 4.9 Data warehouse table design

4.4.2.2 Query for data warehouse:

4.4.2.2.1 Query from administrator for analytics:

4.4.2.2.1.1 Category with most sale by month in a period of time:

Parameter	DATETIME from DATETIME to															
Result description	Return Category with most sale by month															
Query	<pre>SELECT d.month_name, d.year, f.category_id, c.categoryName, SUM(quantity) AS sale FROM fact_order AS f JOIN dim_category AS c JOIN dim_date AS d ON f.product_id = c.category_id AND f.date_id = d.date_id WHERE f.date_id > GetDateKey(from) AND f.date_id < GetDateKey(to) GROUP BY d.month_name ORDER BY sale DESC</pre>															
Example Result	<table border="1"> <thead> <tr> <th>Month_name</th> <th>Year</th> <th>category_id</th> <th>categoryName</th> <th>sale</th> </tr> </thead> <tbody> <tr> <td>Tháng Mười Một</td> <td>2014</td> <td>1</td> <td>Áo sơ mi</td> <td>30</td> </tr> <tr> <td>Tháng Mười Hai</td> <td>2014</td> <td>1</td> <td>Áo Polo</td> <td>16</td> </tr> </tbody> </table>	Month_name	Year	category_id	categoryName	sale	Tháng Mười Một	2014	1	Áo sơ mi	30	Tháng Mười Hai	2014	1	Áo Polo	16
Month_name	Year	category_id	categoryName	sale												
Tháng Mười Một	2014	1	Áo sơ mi	30												
Tháng Mười Hai	2014	1	Áo Polo	16												

4.4.2.2.1.2 Top product sale in a month of an year:

Parameter	INT month_of_year INT year						
Result description	Return top product sale of a month of a year						
Query	<pre>SELECT f.product_id, productName, SUM(quantity) AS sale FROM fact_order AS f JOIN dim_product AS p JOIN dim_date AS d ON f.product_id = p.product_id AND f.date_id = d.date_id WHERE d.month_of_year = month_of_year and d.year = year GROUP BY f.product_id ORDER BY sale DESC</pre>						
Example Result	<table border="1"> <thead> <tr> <th>product_id</th> <th>productName</th> <th>sale</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>Áo sơ mi kẻ sọc</td> <td>30</td> </tr> </tbody> </table>	product_id	productName	sale	1	Áo sơ mi kẻ sọc	30
product_id	productName	sale					
1	Áo sơ mi kẻ sọc	30					

4.4.2.2.1.3 Top store sale by month in a year:

Parameter	INT year
Result description	Return the store which has most sale of all month in a year
Query	<pre>SELECT d.month_name, f.store_id, s.store_name, SUM(quantity) AS sale FROM fact_order AS f JOIN dim_store AS s JOIN dim_date AS d ON f.store_id = s.store_id AND f.date_id = d.date_id WHERE d.year = 2014 GROUP BY d.month_name ORDER BY sale DESC</pre>

Example Result	month_name	store_id	store_name	sale
	Tháng Mười Một	2	Hoàng Hà	30
	Tháng Mười Hai	2	Hoàng Hà	16

4.4.2.2.1.4 Revenue by month in a year:

Parameter	INT year		
Result description	Return revenue of the system of an year		
Query	<pre>SELECT d.month_name, d.year, SUM(total) AS revenue FROM fact_order AS f JOIN dim_date AS d ON f.date_id = d.date_id WHERE d.year = year GROUP BY d.month_name, d.year</pre>		
Example Result	month_name	year	revenue
	Tháng Mười Hai	2014	10047000
	Tháng Mười Một	2014	4500000

4.4.2.2.2 Query for store:

4.4.2.2.2.1 Get most sale and most income product in a month of a year:

Parameter	INT store_id INT month_of_year INT year		
Result description	Get most sale and most income product in a month of a year:		
Query	<pre>SELECT store_name, SUM(quantity) AS sale, SUM(total) AS total FROM fact_order AS f JOIN dim_date AS d JOIN dim_store AS s ON f.date_id = d.date_id AND f.store_id = s.store_id WHERE f.store_id = store_id AND d.month_of_year = month_of_year AND d.year = year</pre>		
Example Result	store_name	sale	total
	Hoàng Hà	13	6907000

4.4.2.2.3 Table detailed design:

4.4.2.2.3.1 Table dim_category

Column	Type	Is Nullable?	Default
<i>category_id</i>	int(10)	No	0
categoryName	varchar(45)	No	
typeName	varchar(45)	No	
targetName	varchar(20)	No	

4.4.2.2.3.2 Table dim_date

Column	Type	Is Nullable?	Default
<i>date_id</i>	int(10)	No	
full_date	date	Yes	NULL
date_name	char(11)	No	
date_name_vn	char(11)	No	
day_of_week	tinyint(4)	No	
day_name_of_week_vn	char(10)	No	
day_name_of_week_en	char(10)	No	
day_of_month	tinyint(4)	No	
day_of_year	smallint(6)	No	
weekday_weekend	char(10)	No	
week_of_year	tinyint(4)	No	
month_name	char(15)	No	
month_name_short	char(10)	No	
month_name_en	char(10)	No	
month_of_year	tinyint(4)	No	
is_last_day_of_month	char(1)	No	
Quarter	tinyint(4)	No	
Year	smallint(6)	No	

4.4.2.2.3.3 Table dim_produc

Column	Type	Is Nullable?	Default
<i>Product_id</i>	int(10)	No	
ProductName	varchar(70)	No	
Price	Double	Yes	NULL
Description	Text	Yes	NULL

4.4.2.2.3.4 *Table dim_store*

Column	Type	Is Nullable?	Default
<i>store_id</i>	int(10)	No	
store_name	varchar(70)	No	
store_address	varchar(100)	Yes	NULL
store_city	varchar(45)	Yes	NULL

4.4.2.2.3.5 *Table dim_user*

Column	Type	Is Nullable?	Default
<i>user_id</i>	int(10)	No	
user_username	varchar(45)	No	
user_address	varchar(100)	Yes	NULL
user_city	varchar(45)	Yes	NULL
user_dob	datetime	Yes	NULL
user_male	bit(1)	Yes	NULL
user_female	bit(1)	Yes	NULL

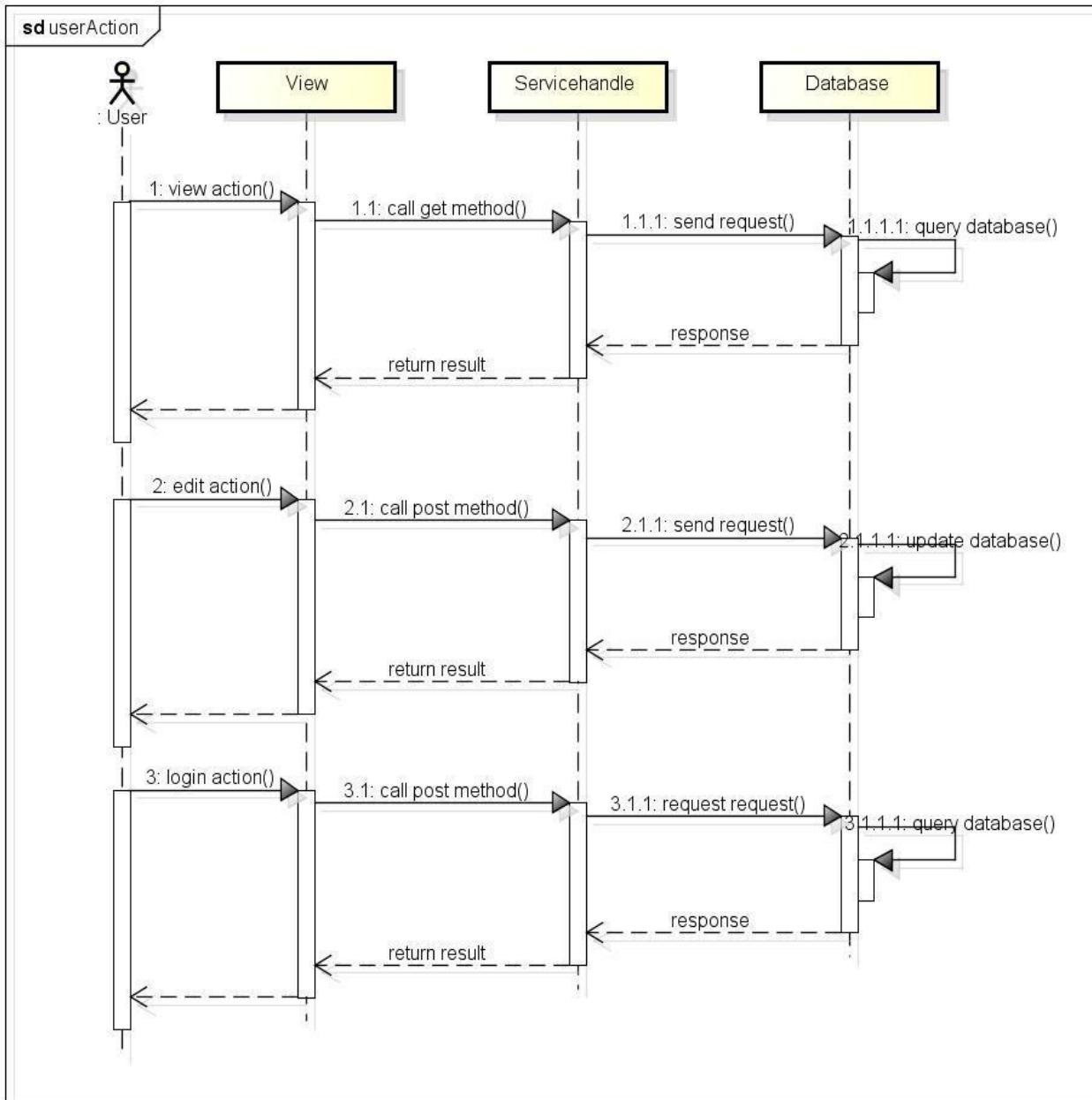
4.4.2.2.3.6 *Table fact_order*

Column	Type	Is Nullable?	Default
user_id	int(10)	No	
product_id	int(10)	No	
store_id	int(10)	No	
date_id	int(11)	No	
category_id	int(10)	No	
quantity	smallint(5)	No	
color	varchar(45)	Yes	NULL
size	varchar(10)	Yes	NULL
price	Double	Yes	NULL
total	Double	Yes	NULL

4.5 Operational Scenarios:

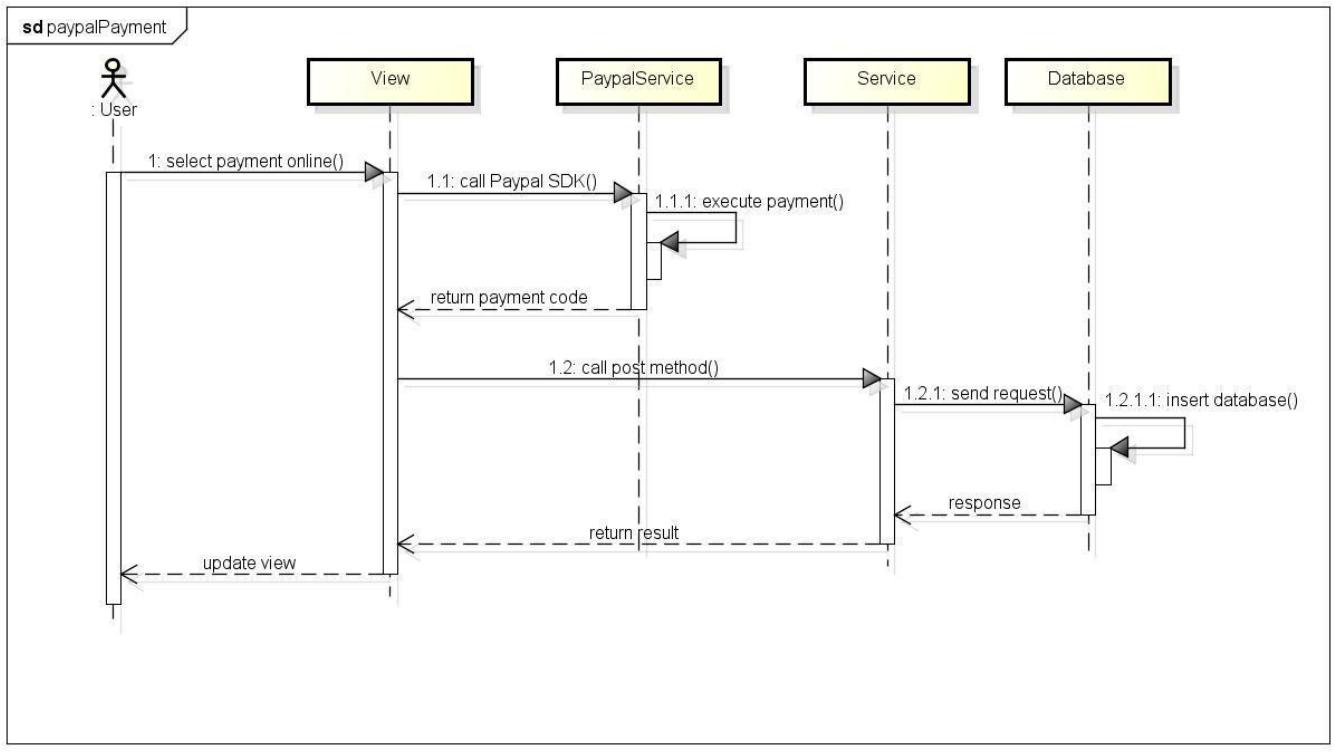
4.5.1 Client Operation:

4.5.1.1 SEND Http GET and POST request to application server:



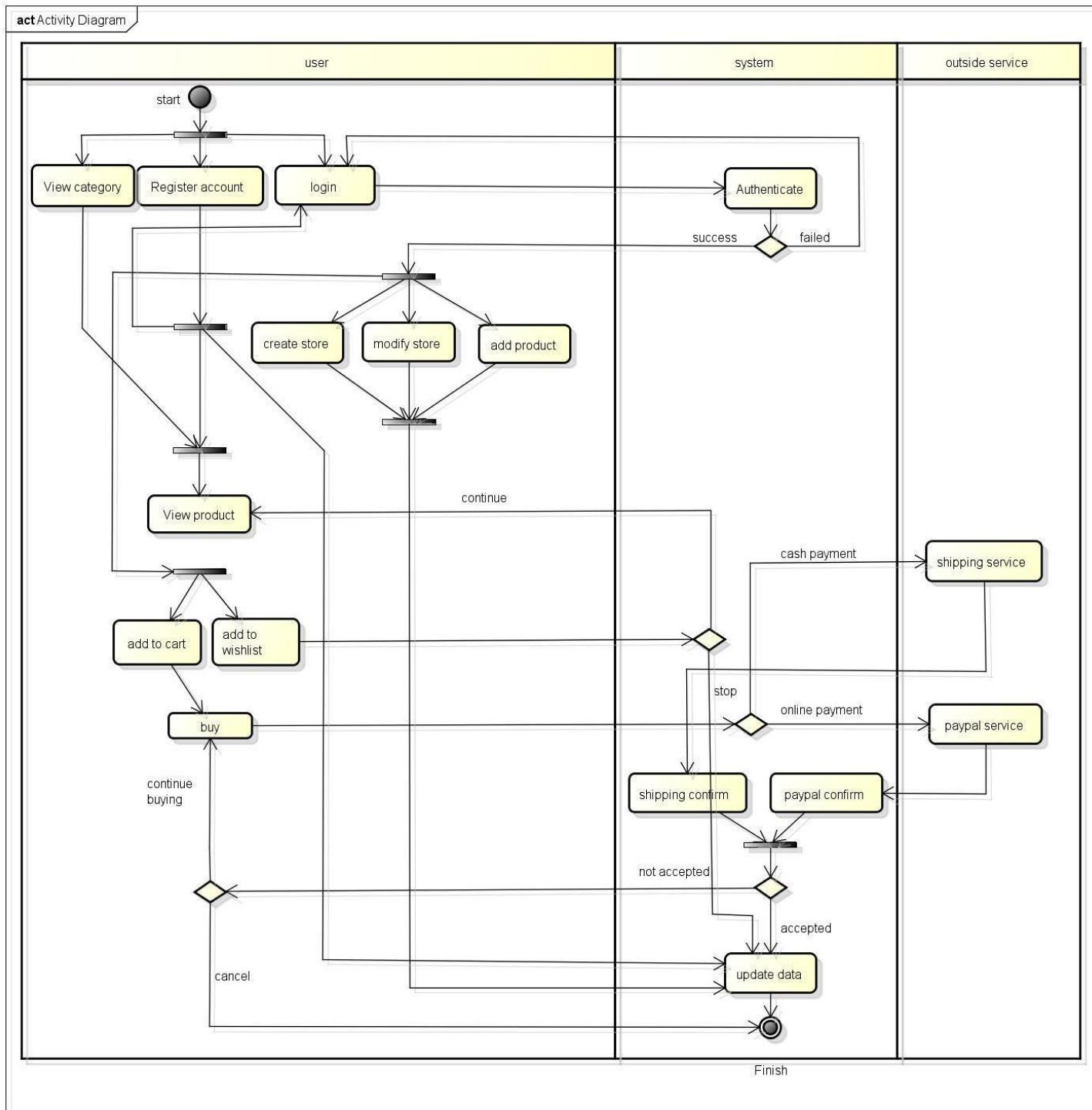
powered by Astah

4.5.1.2 Interaction with Paypal service:



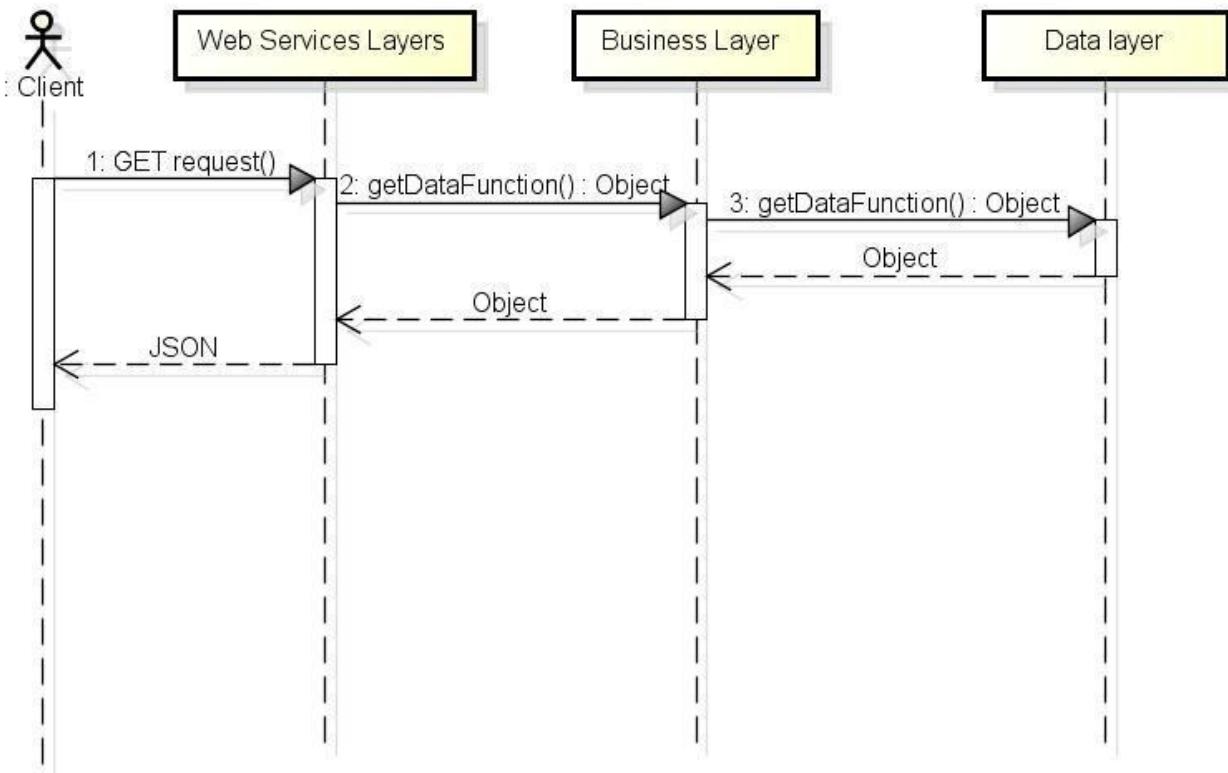
powered by Astah

4.5.1.3 Business process activity diagram:



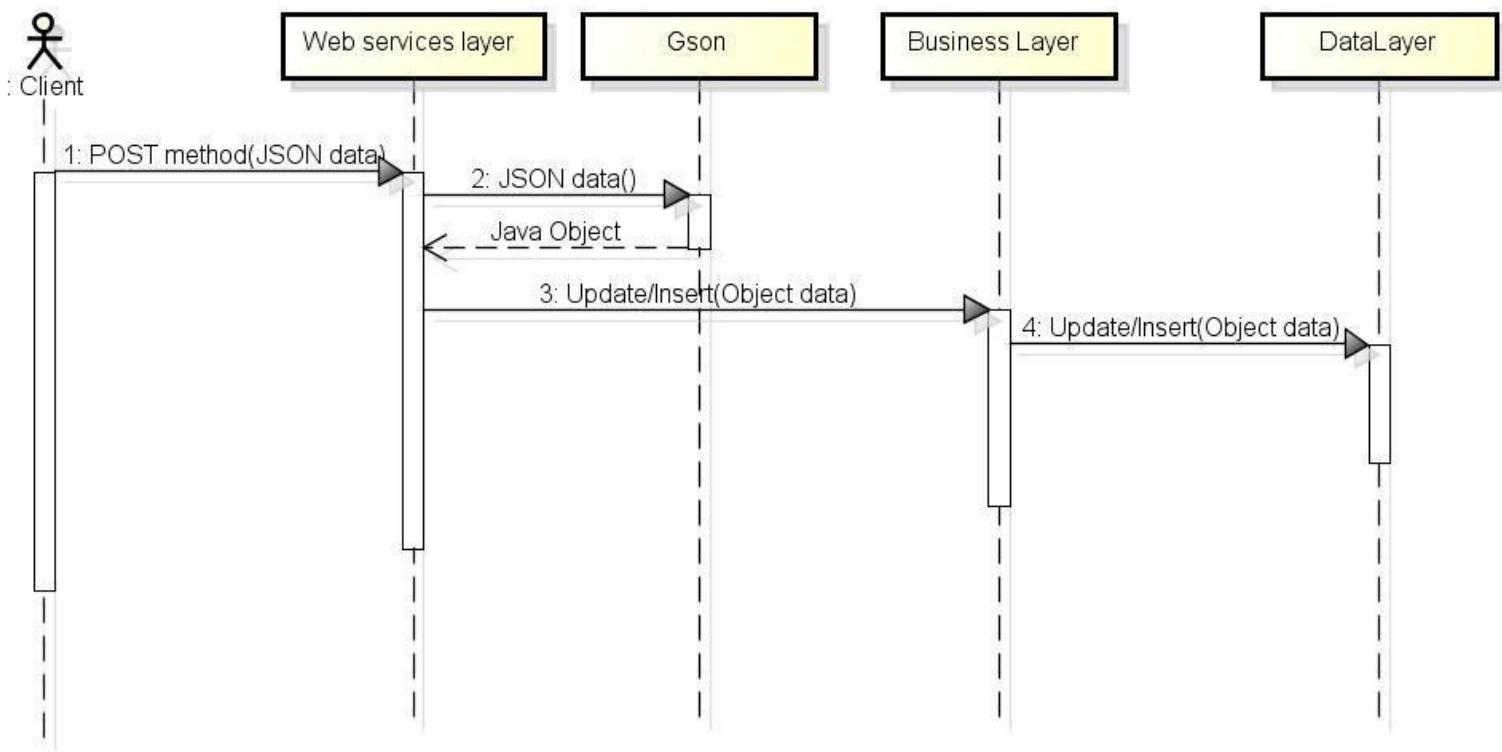
4.5.2 Server Operation:

4.5.2.1 Handle GET request:



powered by Astah

4.5.2.2 Handle POST request:

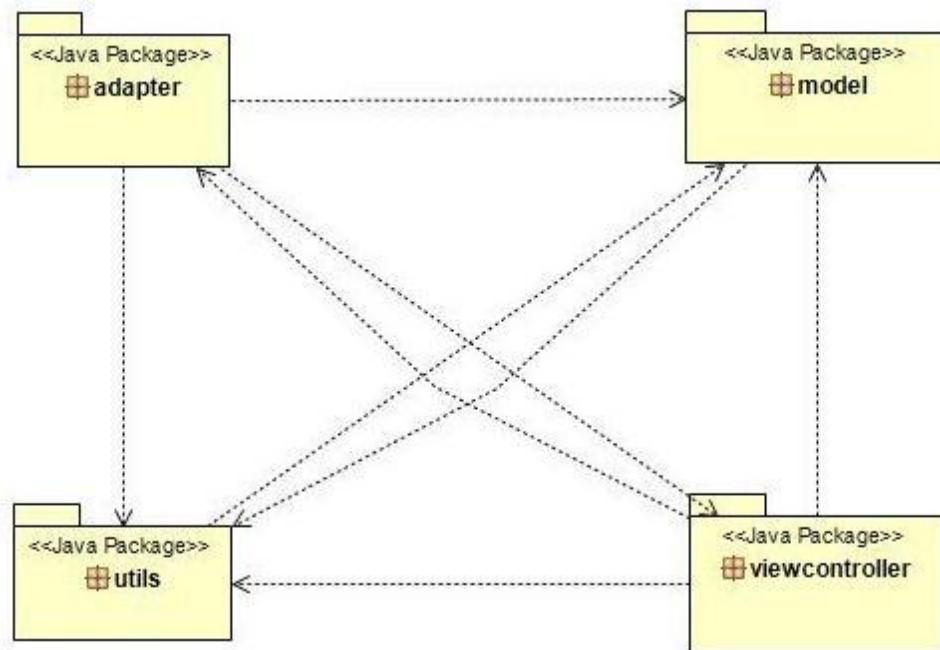


powered by Astah

4.6 Detailed Design

4.6.1 Client Design:

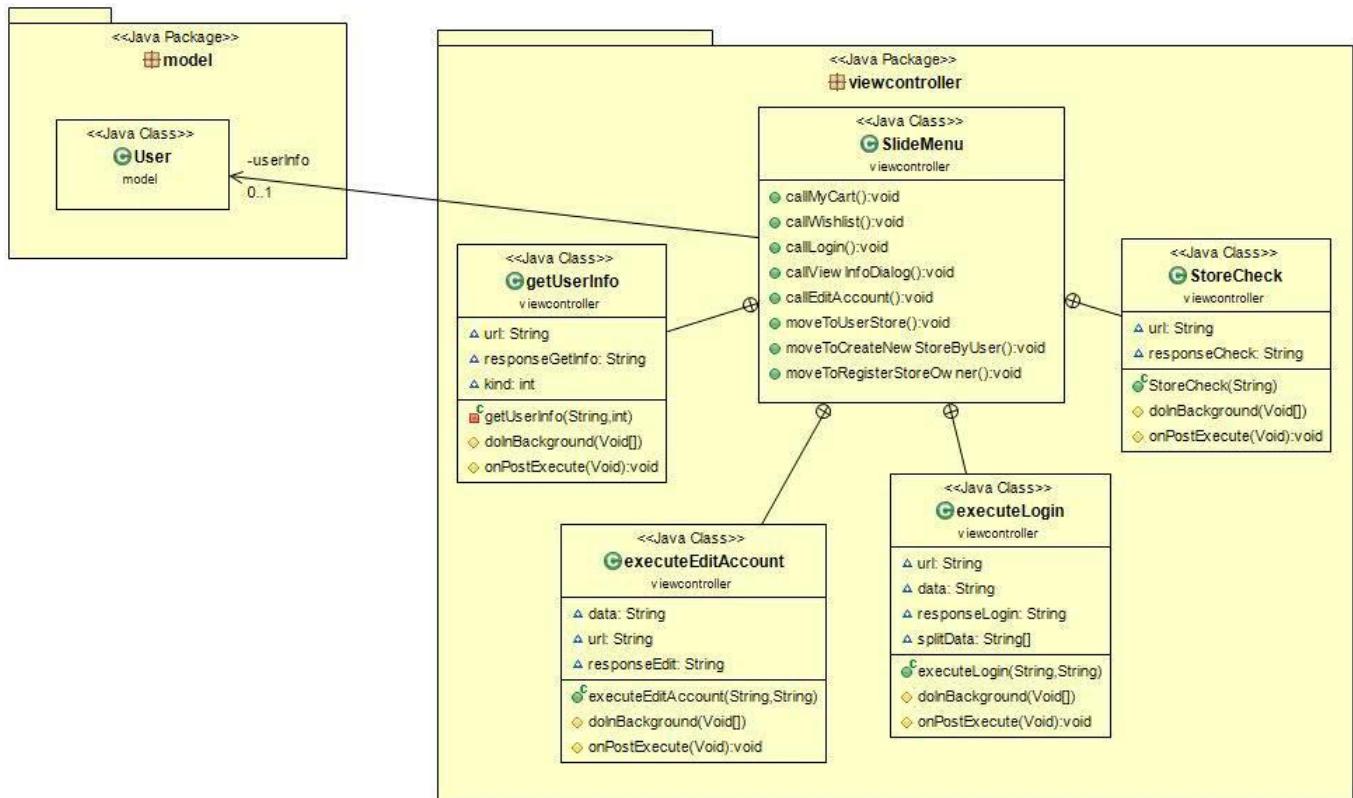
4.6.1.1 Client Package Diagram



4.6.1.2 Client Class Diagram:

4.6.1.2.1 User Operation Class Diagram:

4.6.1.2.1.1 Overview:



4.6.1.2.1.2 Class Detail:

Detail Design:

callLogin()						
public void callLogin()						
Description	Call login dialog.					
Parameter	Name	Type	Description			
	N/A	N/A	N/A			
Return	N/A					
executeLogin()						
public class executeLogin(String url, String data)						
Description	Package username and password. Call post method from service handle and send request to server. Receive and execute response.					
Parameter	Name	Type	Description			
	urlLogin	String	Webservice url			
	data	String	Json string include username and password			
Return	response = [-1,-1] : Login successful. response = [userID, storeOwner] : Login successful.					
callViewInfoDialog()						
public void callViewInfoDialog()						
Description	Call view user information dialog					
Parameter	Name	Type	Description			
	N/A	N/A	N/A			
Return	N/A					
getUserInfo()						
public class getUserInfo(String url, int kind)						
Description	Call get method and send request to server. Get all information of user with userID.					

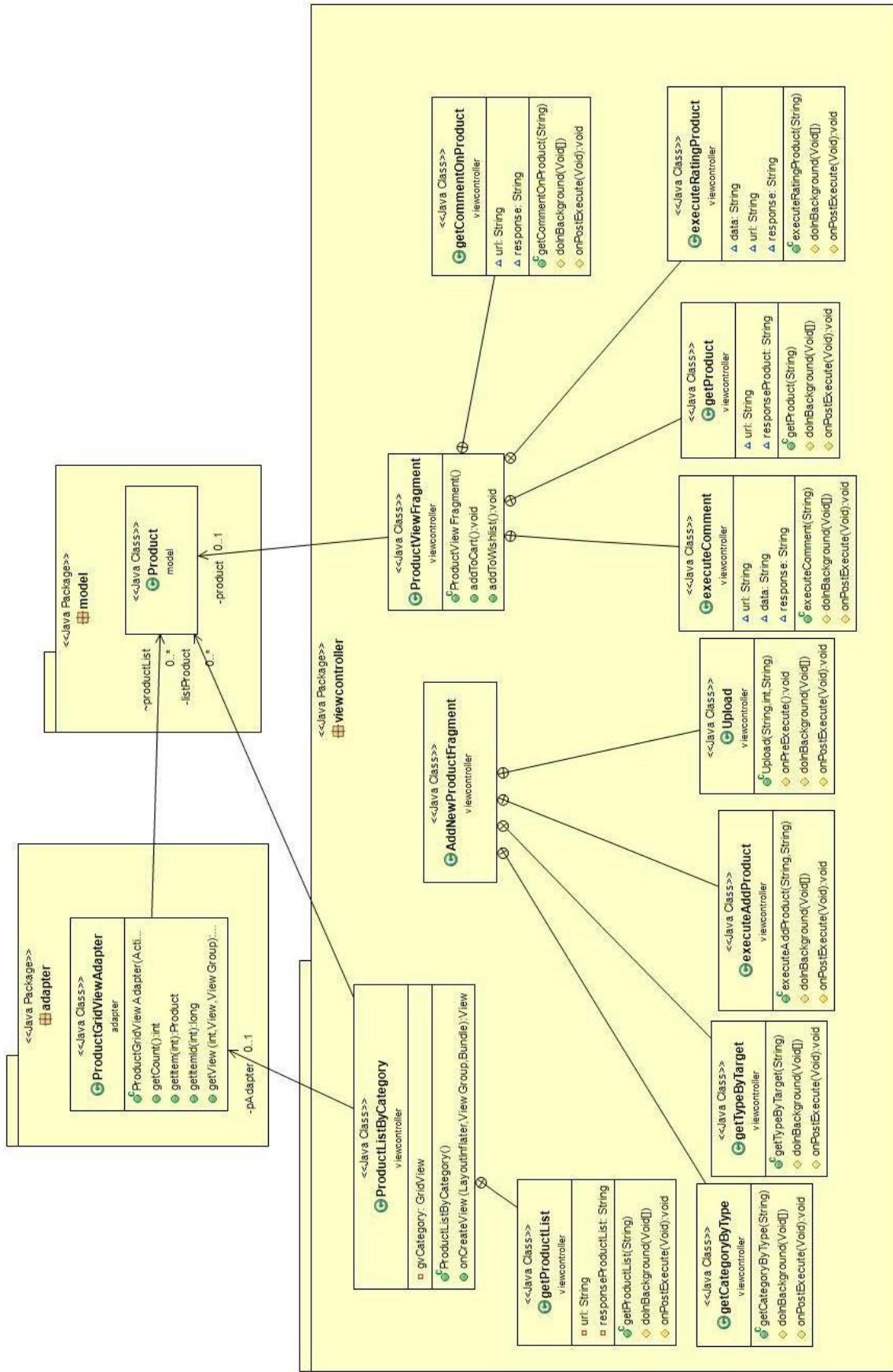
Parameter	Name	Type	Description			
	urlUserInfo	String	Webservice url			
	kind	Int	Purpose of get user information.			
Return	User.					
callEditAccount()						
public void callEditAccount()						
Description	Call edit account dialog.					
Parameter	Name	Type	Description			
	N/A	N/A	N/A			
Return	N/A					
executeEditAccount()						
public class executeEditAccount(String url, String data)						
Description	Package new user's information. Call post method and send request to database. Receive and execute response.					
Parameter	Name	Type	Description			
	urlEditUser	String	Webservice url			
	Data	String	Json string include user object.			
Return	Response = -1 : unsuccessful. Response = userID : successful.					
getCheckStore()						
public void getCheckStore (String url)						
Description	If user is store owner, check User has opened store or not					
Parameter	Name	Type	Description			
	urlCheckStore	String	Webservice url			
Return	null : User does not own store. storeID : User owns store.					
callMyCart()						
public void callMyCart()						

Description	Move to Cart interface.					
Parameter	Name	Type	Description			
	N/A	N/A	N/A			
Return	N/A					
callWishlist()						
public void callWishlist()						
Description	Move to Wishlist interface.					
Parameter	Name	Type	Description			
	N/A	N/A	N/A			
Return	N/A					
moveToUserStore()						
public void moveToUserStore()						
Description	Move to Store interface if user has owned store.					
Parameter	Name	Type	Description			
	N/A	N/A	N/A			
Return	N/A					
moveToCreateNewStoreByUser()						
public void moveToCreateNewStoreByUser()						
Description	Move to register store interface.					
Parameter	Name	Type	Description			
	N/A	N/A	N/A			
Return	N/A					
moveToRegisterStoreOwner()						
public void moveToRegisterStoreOwner()						
Description	Move to register store owner interface.					
Parameter	Name	Type	Description			

	N/A	N/A	N/A
Return	N/A		

4.6.1.2.2 Product Operation Class Diagram:

4.6.1.2.2.1 Overview:



4.6.1.2.2.2 Class Detail:

Detail Design:

getProduct()						
public class getProduct(String url)						
Description	Call get method and send request to server. Receive, execute response and show all information of product.					
Parameter	Name urlProduct	Type String	Description Webservice url			
Return	Type: Product					
getCommentOnProduct()						
public class getCommentOnProduct(String url)						
Description	Call get method and send request to server. Receive, execute response and show all information of product.					
Parameter	Name urlGetComment	Type String	Description Webservice url			
Return	Type: Comment[]					
executeComment()						
public class executeComment(String url, String data)						
Description	Call get method and send request to server. Receive, execute response and show all information of product.					
Parameter	Name urlComment	Type String	Description Webservice url			
	Name data	Type String	Description Json string include comment object			
Return	Response = -1 : unsuccessful. Response = commentID : successful.					
executeRatingProduct()						
public class executeRatingProduct(String url, String data)						
Description	Call get method and send request to server. Receive, execute response and show all information of product.					
Parameter	Name urlRateProduct	Type String	Description Webservice url			

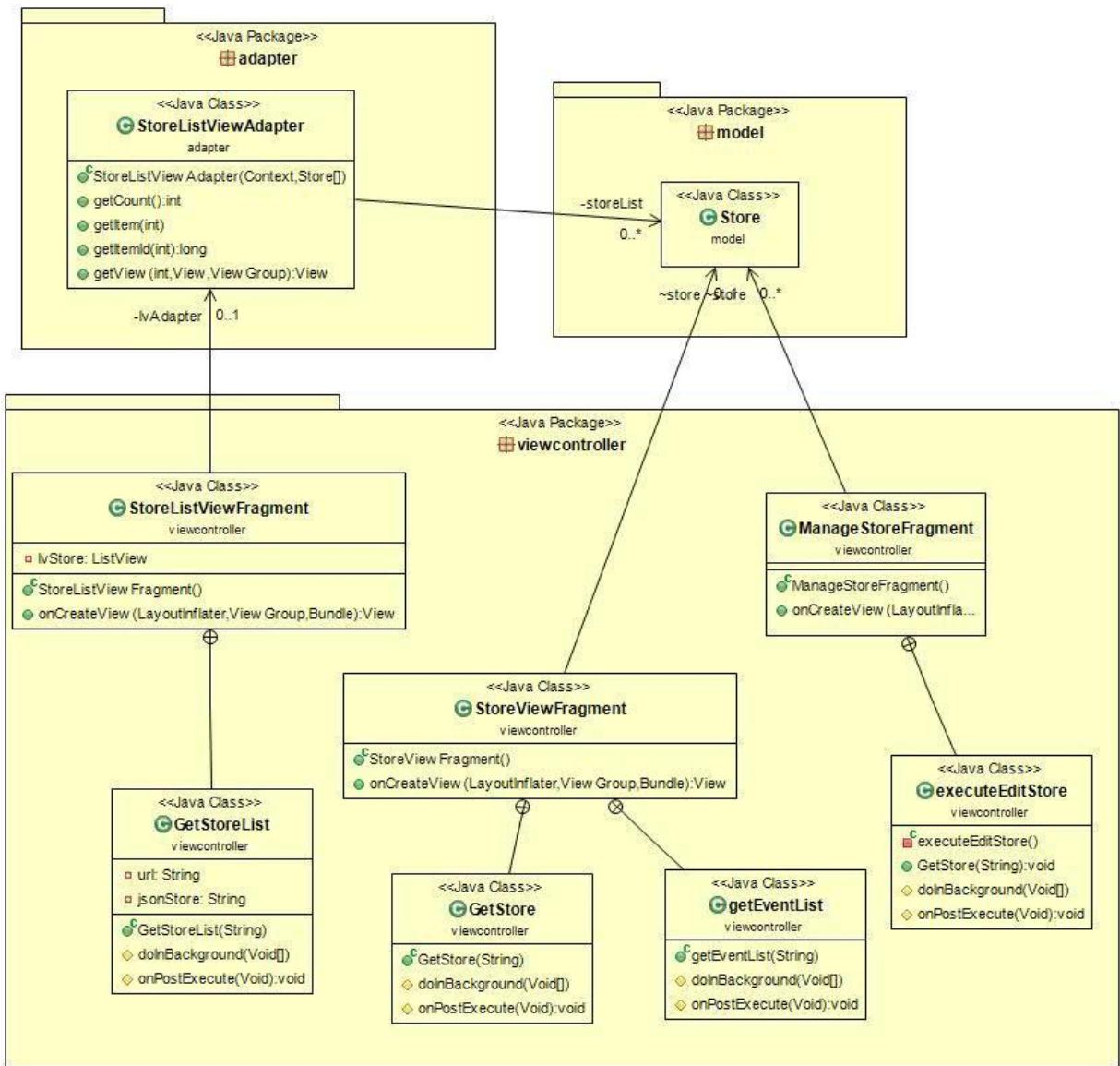
	data	String	Json string include rating product object			
Return	Response = -1 : unsuccessful. Response = 1 : successful.					
addToCart()						
public void addToCart()						
Description	Add product to Cart.					
Parameter	Name	Type	Description			
	N/A	N/A	N/A			
Return	N/A					
addToWishlist()						
public void addToWishlist()						
Description	Add product to Wishlist					
Parameter	Name	Type	Description			
	N/A	N/A	N/A			
Return	N/A					
getTypeByTarget()						
public class getTypeByTarget()						
Description	Get list type product.					
Parameter	Name	Type	Description			
	urlGetType	String	Webservice url			
Return	String[]					
getCategoryByType()						
public class getCategoryByType()						
Description	Get list category product by type.					
Parameter	Name	Type	Description			
	urlGetCategory	String	Webservice url			
Return	String[]					

Upload()						
public class Upload()						
Description	Call upload method Upload image to server from device. Response image url.					
Parameter	Name	Type	Description			
	fileName	String	Image path on device.			
	sequenceNo	Int	Sequence number of image.			
	imagePath	String	Address upload image			
Return	Image url after upload.					
executeAddProduct()						
public class executeAddProduct()						
Description	Packge all product's information. Call post method and send request to server. Receive and execute response.					
Parameter	Name	Type	Description			
	urlAddProuct	String	Webservice url			
	data	String	Json string include all information of product			
Return	Response = -1 : unsuccessful. Response != -1 : successful.					
getProductList()						
public class getProductList()						
Description	Get list product.					
Parameter	Name	Type	Description			
	urlGetProductList	String	Webservice url			
Return	Product[]					

Software Architecture Design

4.6.1.2.3 Store Operation Class Diagram:

4.6.1.2.3.1 Overview:



4.6.1.2.3.2 Class Detail:

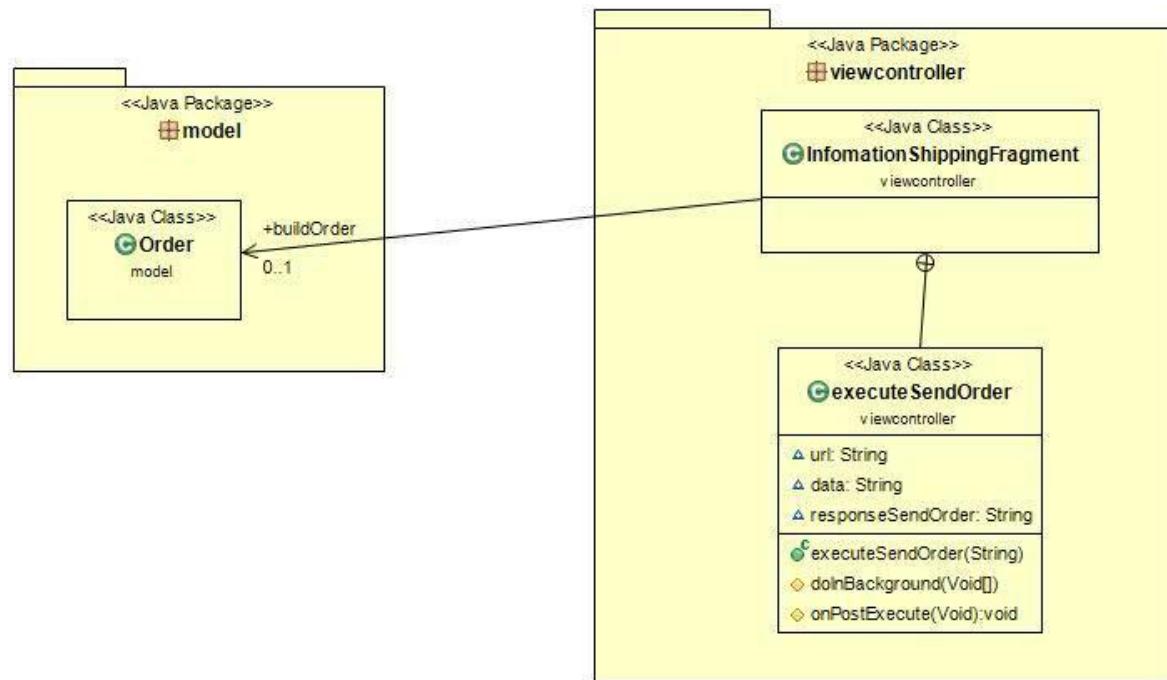
Detail Design:

getStoreList()			
public class getStoreList()			
Description	Get list store.		
Parameter	Name	Type	Description

	urlStoreList	String	Webservice url			
Return	Store[]					
getStore()						
public class getStore()						
Description	Get all information store.					
Parameter	Name urlStore	Type String	Description Webservice url			
Return	Store					
getEventList()						
public class getEventList()						
Description	Get all information store.					
Parameter	Name urlEventList	Type String	Description Webservice url			
Return	utils.Event[]					
executeEditStore()						
public class executeEditStore()						
Description	Package all new and old information of store. Call post method and send request. Receive and execute response.					
Parameter	Name urlEditStore	Type String	Description Webservice url			
Return	Response = -1 : unsuccessful. Response = storeID : successful.					

4.6.1.2.4 Order Operation Class Diagram:

4.6.1.2.4.1 Overview:

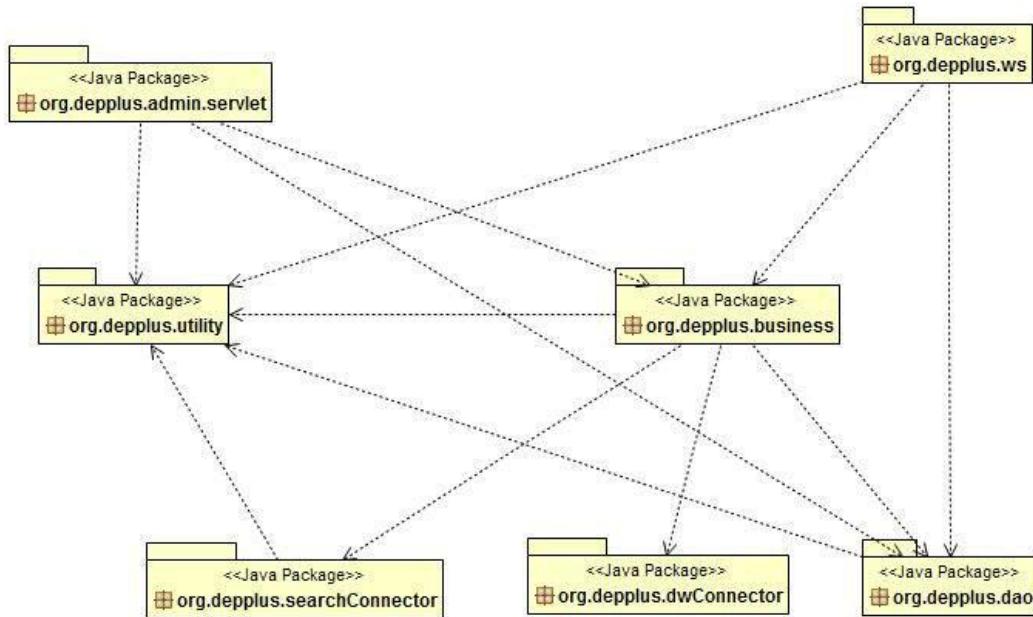


4.6.1.2.4.2 Class Detail:

executeSendOrder			
public class executeSendOrder()			
Description	Package all information of order Call post method and send request. Receive and execute response.		
Parameter	Name	Type	Description
	urlSendOrder	String	Webservice url
Return	Response = -1 : unsuccessful. Response = 1 : successful.		

4.6.2 Application Server Design:

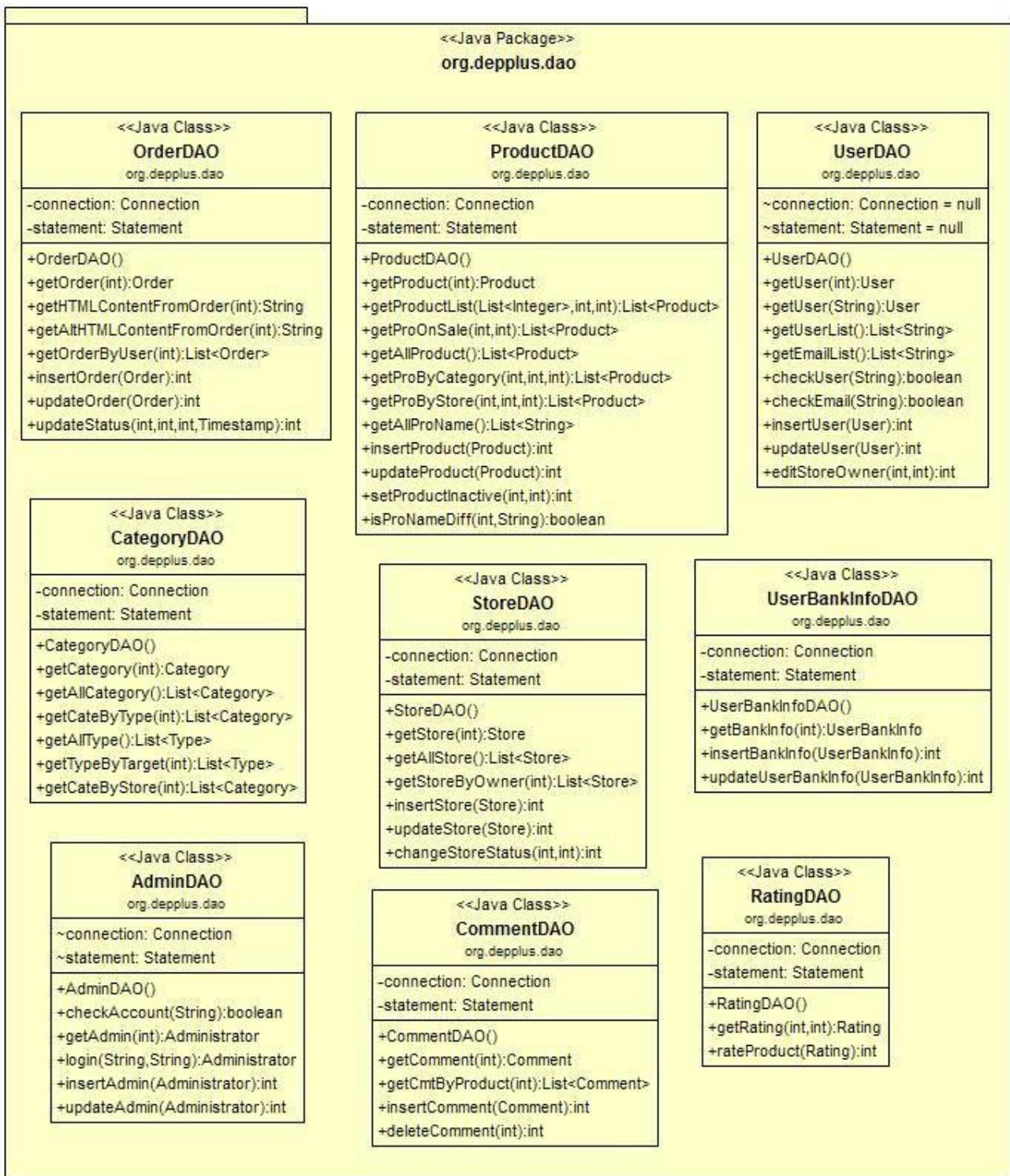
4.6.2.1 Server Package Diagram:



4.6.2.2 Server Class Diagram and detailed design

4.6.2.2.1 Data Access Object package:

Software Architecture Design



4.6.2.2.1.1 AdminDAO class detail:

Method Detail:

checkAccount()						
public boolean checkAccount(java.lang.String account)						
Description	Check account.					
Parameter	Name	Type	Description			
	account	java.lang.String	the account to check			
Return	true, if account not existed, false otherwise					
getAdmin()						
public Administrator getAdmin(int adminID)						
Description	Gets the admin. with the corresponding id					
Parameter	Name	Type	Description			
	adminID	int	the admin id			
Return	the admin with the corresponding id					
Login()						
public Administrator login(java.lang.String account, java.lang.String password)						
Description	Administrator Login.					
Parameter	Name	Type	Description			
	account	java.lang.String	the admin account name			
	password	java.lang.String	the password			
Return	the administrator with id > 0 if success, the blank administrator with id = -1 if fail					
insertAdmin()						
public int insertAdmin(Administrator admin)						
Description	Insert a new admin.					
Parameter	Name	Type	Description			
	admin	Administrator	the admin to insert to database			
Return	the new admin id if success, -1 if fail					
updateAdmin()						

public int updateAdmin/Administrator admin)

public int updateAdmin/Administrator admin)			
Description	Update admin.detail		
Parameter	Name admin	Type Administrator	Description the admin to update
Return	the admin id if success, -1 if fail		

4.6.2.2.1.2 CategoryDAO class detail

Method Detail:

getCategory()

public Category getCategory(int categoryID)

getCategory()			
public Category getCategory(int categoryID)			
Description	Gets the category with corresponding id.		
Parameter	Name categoryID	Type int	Description the category id
Return	the category with the id		

getCategoryByType()

public java.util.List<Category> getCategoryByType(int type)

getCategoryByType()			
public java.util.List<Category> getCategoryByType(int type)			
Description	Gets the category list by type.		
Parameter	Name type	Type int	Description the type of the category list.
Return	the category list by type		

getTypeByTarget()

public java.util.List<Type> getTypeByTarget(int targetID)

getTypeByTarget()			
public java.util.List<Type> getTypeByTarget(int targetID)			
Description	Gets the list of type by a target.		
Parameter	Name targetID	Type int	Description the target id of the type list.
Return	the list of type by a target.		

getCategoryByStore()

public java.util.List<Category> getCatByStore (int storeID)

public java.util.List<Category> getCatByStore (int storeID)			
Description	Gets the category list of products of a store.		
Parameter	Name storeID	Type int	Description the id of a store
Return	the category list of products of a store.		

4.6.2.2.1.3 CommentDAO class detail

Method Detail:

getComment()

public Comment getComment(int commentID)
--

getComment()			
public Comment getComment(int commentID)			
Description	Gets the comment with corresponding id		
Parameter	Name commentID	Type int	Description the comment id
Return	the comment with corresponding id		

getCmtByProduct()

public java.util.List<Comment> getCmtByProduct (int productID)
--

getCmtByProduct()			
public java.util.List<Comment> getCmtByProduct (int productID)			
Description	Gets the comment list of a product.		
Parameter	Name productID	Type int	Description the product id of the product.
Return	the comment list of a product.		

insertComment()

public int insertComment(Comment comment)

insertComment()			
public int insertComment(Comment comment)			
Description	Insert comment.		
Parameter	Name comment	Type Comment	Description the comment to insert
Return	comment id if success, -1 if fail		

deleteComment()

public int deleteComment(int commentID)			
Description	Delete a comment.		
Parameter	Name	Type	Description
	commentID	int	the comment id of the comment
Return	comment id if success, -1 if fail		

Description	Delete a comment.		
Parameter	Name	Type	Description
	commentID	int	the comment id of the comment
Return	comment id if success, -1 if fail		

4.6.2.2.1.4 EventDAO class detail

Method Detail:

getEvent()			
public Event getEvent(int eventID)			
Description	Gets the event with corresponding id.		
Parameter	**Name**	**Type**	**Description**
eventID	int	the event id	
Return	the event		
getAllEvent()			
public java.util.List<Event> getAllEvent()			
Description	Get all events.		
Parameter	**Name**	**Type**	**Description**
N/A	N/A	N/A	
Return	list of all events		
getEventByStore()			
public java.util.List<Event> getEventByStore(int storeID)			
Description	Gets the event by store.		
Parameter	**Name**	**Type**	**Description**
storeID	int	the store id	
Return	the event by storeID		
getEventProduct()			

public java.util.List<Product> getEventProduct(int eventID, int page, int num)			
Description	Gets the product that is on event		
Parameter	Name	Type	Description
	eventID	int	the event id
	page	int	the page
num	int	the number per page	
Return	the product list is on event		

4.6.2.2.1.5 OrderDAO class detail

Method Detail:

getOrder()						
public Order getOrder(int orderID)						
Description	Gets the order with corresponding id from database.					
Parameter	Name	Type	Description			
	orderID	int	the order id to match.			
Return	the order with corresponding id					
getOrderByUser()						
public java.util.List<Order> getOrderByUser(int userID)						
Description	Gets the order by user.					
Parameter	Name	Type	Description			
	userID	int	the user id			
Return	the order by user					
insertOrder()						
public int insertOrder(Order order)						
Description	Insert order.					
Parameter	Name	Type	Description			
	order	Order	the order			
Return	the int					

updateOrder()						
public int updateOrder(Order order)						
Description	Update order.					
Parameter	Name order	Type Order	Description the order			
Return	the int					
updateStatus()						
public int updateStatus (int orderID, int adminID, int statusType, java.sql.Timestamp time)						
Description	Update status.					
Parameter	Name orderID adminID statusType time	Type int int int java.sql.Timestamp	Description the order id the admin id the status type the time			
Return	the int					

4.6.2.2.1.6 ProductDAO class detail

Method Detail:

getProduct()						
public Product getProduct(int productID)						
Description	Gets the product by the id.					
Parameter	Name productID	Type int	Description the id of the product to get			
Return	the product with corresponding id					
getEventProduct()						
public java.util.List<Product> getEventProduct(int eventID, int page, int num)						
Description	Gets the list of products which id is in a list.					
Parameter	Name	Type	Description			

	idlist	java.util.List<java.lang.Integer>	the list of id			
	page	int	the page of the product list			
	num	int	the number of products per page			
Return	the product list					
getProByCategory()						
public java.util.List<Product> getProByCategory(int categoryID, int page,int num)						
Description	Gets the product list by category id.					
Parameter	Name	Type	Description			
	categoryID	int	the category id of the product inside the list			
	page	int	the page of the list			
	num	int	the number of products per page			
Return	the pro by category					
getProByStore						
public java.util.List<Product> getProByStore(int storeID, int page, int num)						
Description	Gets the product by store.					
Parameter	Name	Type	Description			
	storeID	int	the store id of the product list			
	page	int	the page of the product list			
	num	int	the number or product per list			
Return	the pro by store					
insertProduct()						
public int insertProduct(Product product)						
Description	Insert product.					
Parameter	Name	Type	Description			
	product	Product	the product to insert			
Return	the product id if success, -1 if fail					

updateProduct()						
public int updateProduct(Product product)						
Description	Update product.					
Parameter	Name product	Type Product	Description the product to update			
Return	the product id if success, -1 if fail					
setProductInactive()						
public int setProductInactive(int productID, int adminID)						
Description	Sets the product inactive.					
Parameter	Name productID	Type int	Description the id of the product.			
	Name adminID	Type int	Description the id of the administrator who commit			
Return	the 1 if success, -1 if fail					
isProNameDiff()						
public boolean isProNameDiff(int proID, java.lang.String proName)						
Description	Checks if the name is different from the product name in database.					
Parameter	Name proID	Type int	Description the product id			
	Name proName	Type java.lang.String	Description the new product name			
Return	true, if is pro name is different					

4.6.2.2.1.7 RatingDAO class detail

Method Detail:

getRating()			
public Rating getRating(int productID,int userID)			
Description	Gets the rating of a product by an user.		
Parameter	Name	Type	Description

	productID	int	the id of the product			
	userID	int	the id of the user			
Return	the rating of a product by an user					
rateProduct()						
public int rateProduct(Rating rating)						
Description	Rate a product.					
Parameter	Name	Type	Description			
	rating	Rating	the rating of the product			
Return	the 1 if success, -1 if fail					

4.6.2.2.1.8 StoreDAO

Method Detail:

getStore()						
public Store getStore(int storeID)						
Description	Gets the store with corresponding id.					
Parameter	Name	Type	Description			
	storeID	int	the id of the store			
Return	the store with the corresponding id.					
getAllStore()						
public java.util.List<Store> getAllStore()						
Description	Gets a list of all store in database					
Parameter	Name	Type	Description			
	N/A	N/A	N/A			
Return	A list of all store in database					
getStoreByOwner()						
public java.util.List<Store> getStoreByOwner(int userID)						
Description	Gets the store by owner.with an userid					
Parameter	Name	Type	Description			

	userID	int	the id of owner			
Return	the store by owner.with an userid					
insertStore()						
public int insertStore(Store store)						
Description	Insert store.					
Parameter	Name	Type	Description			
	store	Store	the store to insert			
Return	the store id if success, -1 if fail					
updateStore()						
public int updateStore (Store store)						
Description	Update store.					
Parameter	Name	Type	Description			
	store	Store	the store info to update			
Return	the store id if success, -1 if fail					
changeStoreStatus()						
public int changeStoreStatus(int storeID,int adminID)						
Description	Change store status.					
Parameter	Name	Type	Description			
	storeID	int	the store id			
	adminID	int	the admin id			
Return	the int					

4.6.2.2.1.9 UserDAO

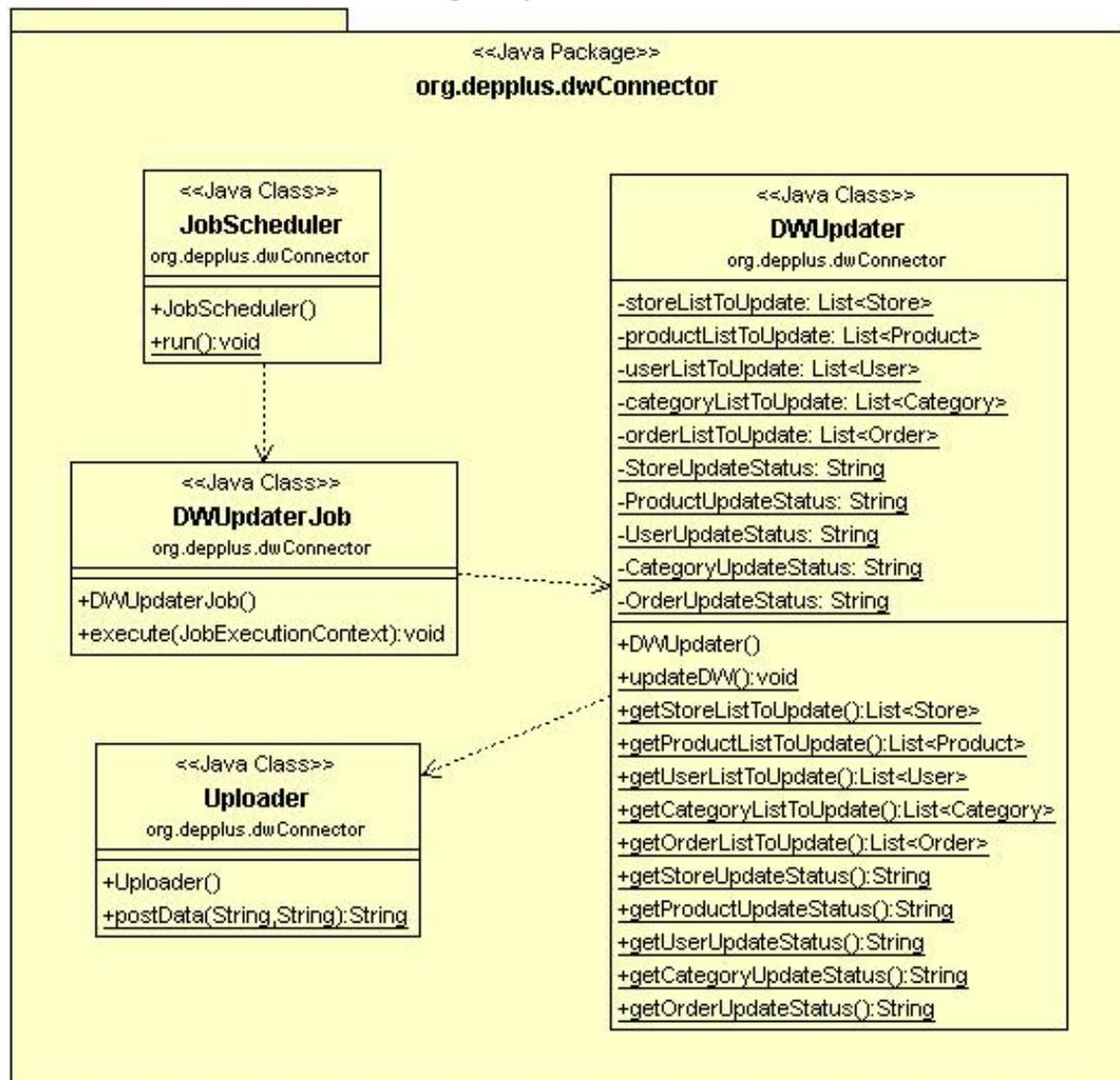
Method Detail:

getUser()			
public User getUser(int userid)			
Description	Gets the user with the corresponding id.		
Parameter	Name	Type	Description

	userid	int	the id of the user			
Return	the user with the corresponding id.					
getUser()						
public User getUser(java.lang.String username)						
Description	Gets the user.					
Parameter	Name username	Type java.lang.String	Description the username of the user			
Return	the user with corresponding username					
checkUser()						
public boolean checkUser(java.lang.String username)						
Description	Check if username existed in database.					
Parameter	Name username	Type java.lang.String	Description the username to check			
Return	true, if not existed, false otherwise					
checkEmail()						
public boolean checkEmail(java.lang.String email)						
Description	Check if email existed.					
Parameter	Name email	Type java.lang.String	Description the email to check			
Return	true, if successful, false otherwise					
insertUser()						
public int insertUser(User user)						
Description	Insert an user to database.					
Parameter	Name user	Type User	Description the user to insert			
Return	the user id if success, -1 if fail					
updateUser()						

public int updateUser(User user)						
Description	Update an user.					
Parameter	Name	Type	Description			
	user	User	the user			
Return	the user id if success, -1 if fail					
grantStoreOwner()						
public int grantStoreOwner(int userID, int action)						
Description	edit store owner status.					
Parameter	Name	Type	Description			
	userID	int	the user id			
	action	int	the action			
Return	1 if success, -1 if fail					

4.6.2.2.2 Data warehouse connector package:



4.6.2.2.3 Search Engine Connector:



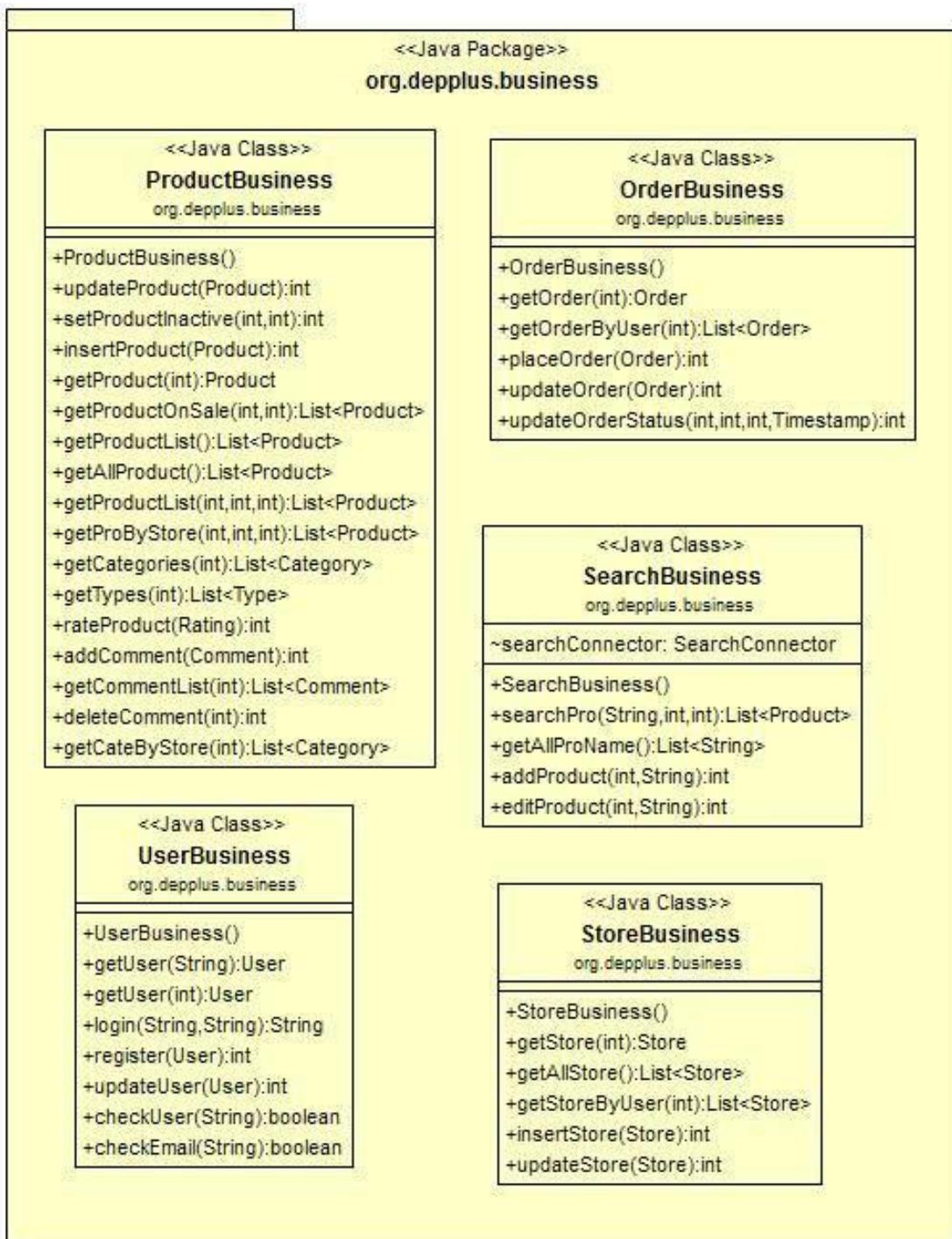
4.6.2.2.3.1 SearchConnector class detail:

Method Detail:

getIdFromSearch()						
public java.util.List<java.lang.Integer> getIdFromSearch(java.lang.String query, int page, int num)						
Description	Gets the product id list from search Engine.					
Parameter	Name	Type	Description			
	query	java.lang.String	the query to search			
	page	int	the page of product list			
	num	int	the number of product per page			
Return	the id list of product returned from search engine					
addProduct()						
public int addProduct(int proID,java.lang.String proName)						
Description	Adds the product to search datasource.					
Parameter	Name	Type	Description			
	proID	int	the id of the product			
	proName	java.lang.String	the name of the			

			product			
Return	1 if success, -1 if fail					
editProduct()						
public int editProduct(int proID,java.lang.String proName)						
Description	Edit product to search datasource.					
Parameter	Name	Type	Description			
	proID	int	the id of the product			
	proName	java.lang.String	the name of the product			
Return	1 if success, -1 if fail					

4.6.2.2.4 Business package:



4.6.2.2.4.1 EventBusiness Class Detail:

Method list:

getEvent()						
public Event getEvent(int eventID)						
Description	Gets the event with corresponding id					
Parameter	Name	Type	Description			
	eventID	int	the event id			
Return	the event with corresponding id					
getAllEvent()						
List<Event> getAllEvent()						
Description	Get all available event					
Parameter	Name	Type	Description			
Return	List of all available event					
getEventByStore()						
public java.util.List<Event> getEventByStore(int storeID)						
Description	Gets the event by store.					
Parameter	Name	Type	Description			
	storeID	int	the store id			
Return	the event of a store					
getEventProduct()						
public java.util.List<Product> getEventProduct(int eventID)						
Description	Gets the list product of an event					
Parameter	Name	Type	Description			
	eventID	int	the event id			
Return	List of products on the event					

4.6.2.2.4.2 OrderBusiness Class Detail

Method Detail:

getOrder()						
public Order getOrder(int orderID)						
Description	Gets the order with the corresponding id					
Parameter	Name	Type	Description			
	orderID	int	ID of the order need to get			
Return	the order with matched id, -1 if not found					
getOrderByUser()						
public java.util.List<Order> getOrderByUser(int userID)						
Description	Get the list of the order placed by the user with an id					
Parameter	Name	Type	Description			
	userID	Int	The id of the user			
Return	A list of the order placed by the user with an id					
placeOrder()						
public int placeOrder(Order order)						
Description	Place a new order.					
Parameter	Name	Type	Description			
	order	Order	the order to be added			
Return	the ID of the new order if success, -1 if fail					
updateOrder()						
public int updateOrder(Order order)						
Description	Update an existing order.					
Parameter	Name	Type	Description			

	order	Order	the order to be updated			
Return	the id of the updated order if success, -1 if fail					
updateOrderStatus()						
public int updateOrderStatus(int orderID, int adminID,int statusType,java.sql.Timestamp time)						
Description	Update order status.					
Parameter	Name	Type	Description			
	orderID	int	the order id			
	adminID	int	the administrator id			
	statusType	int	the status type			
	time	java.sql.Timestamp	the time the update action occur			
Return	the number of status changed, -1 if error					

4.6.2.2.4.3 ProductBusiness Class Detail

Method Detail:

updateProduct()						
public int updateProduct(Product product)						
Description	Update product.					
Parameter	Name	Type	Description			
	product	Product	the product needed to be updated			
Return	the id of the product if success, -1 if fail					
setProductInactive()						
public int setProductInactive(int productID,int adminID)						
Description	Sets the product to inactive.					
Parameter	Name	Type	Description			
	productID	int	the product id			
	adminID	int	the id of the administrator who commit			

Return	1 if success, 0 if fail, -1 if error					
insertProduct()						
public int insertProduct(Product product)						
Description	Insert new product.					
Parameter	Name	Type	Description			
	product	Product	the product to insert			
Return	the the ID of the new product if success, -1 if fail					
getProduct()						
public Product getProduct(int productID)						
Description	Gets the product with the corresponding ID.					
Parameter	Name	Type	Description			
	productID	int	the product id			
Return	the product with matched id					
getProByCate()						
public java.util.List<Product> getProByCate(int categoryID, int page, int num)						
Description	Gets the list of products in a category.					
Parameter	Name	Type	Description			
	categoryID	int	the category id			
	page	int	the page of the product list			
	num	int	the number of products per list			
Return	the product list					
getProByStore()						
public java.util.List<Product> getProByStore(int storeID, int page, int num)						
Description	Gets the product by store.					
Parameter	Name	Type	Description			
	storeID	int	the store id			
	page	int	the page of the list			

	num	int	the number of product per list			
Return	the product list by of a store					
getCategories()						
public java.util.List<Category> getCategories(int typeID)						
Description	Gets the categories.					
Parameter	Name	Type	Description			
	typeID	int	the type id			
Return	the list categories belong to the type					
getTypes()						
public java.util.List<Type> getTypes(int targetID)						
Description	Gets the types.					
Parameter	Name	Type	Description			
	targetID	int	the target id			
Return	the types belong to a target					
rateProduct()						
public int rateProduct(Rating rating)						
Description	Rate a product.					
Parameter	Name	Type	Description			
	rating	Rating	the rating of the product			
Return	1 if success, -1 if fail					
addComment()						
public int addComment(Comment comment)						
Description	Adds the comment.					
Parameter	Name	Type	Description			
	comment	Comment	the comment			
Return	comment id if success, -1 if fail					

getCommentList()

public java.util.List<Comment> getCommentList(int productID)

Description	Gets the list of comments of a product.		
Parameter	Name	Type	Description
	productID	int	the product id
Return	the list of comments of a product.		

deleteComment()

public int deleteComment(int commentID)

Description	Delete a comment.		
Parameter	Name	Type	Description
	commentID	int	the comment id
Return	comment id if success, -1 if fail		

getCategoryByStore()

public java.util.List<Category> getCategoryByStore(int storeID)

Description	Gets the category list by store.		
Parameter	Name	Type	Description
	storeID	int	the id of the store
Return	the category list of a store		

4.6.2.2.4.4 SearchBusiness Class detail

Method Detail

searchPro()

public java.util.List<Product> searchPro(java.lang.String query, int page, int num)

Description	Search product.		
Parameter	Name	Type	Description
	query	java.lang.String	the search query string
	page	int	the page of product list

			result			
	num	int	the number per page			
Return	the list of product whose name matched with the query					
addProduct()						
public int addProduct(int proID,java.lang.String proName)						
Description	Adds the product to the search engine.					
Parameter	Name	Type	Description			
	proID	int	the pro id			
	proName	java.lang.String	the pro name			
	Return					
1 if success, -1 if fail						
editProduct()						
public int editProduct(int proID,java.lang.String proName)						
Description	Adds the product to the search engine.					
Parameter	Name	Type	Description			
	proID	int	the pro id			
	proName	java.lang.String	the pro name			
	Return					
1 if success, -1 if fail						

4.6.2.2.4.5 StoreBusiness Class detail

Method Detail:

getStore()			
public Store getStore(int storeID)			
Description	Gets the store with corresponding id.		
Parameter	Name	Type	Description
	storeID	int	the id of the store
Return	the store with the corresponding id		

getStore()						
public java.util.List<Store> getAllStore()						
Description	Get a list all available store					
Parameter	Name	Type	Description			
	N/A	N/A	N/A			
Return	A list all the available store					
getStoreByUser()						
public java.util.List<Store> getStoreByUser(int userID)						
Description	Gets the store owned by user.					
Parameter	Name	Type	Description			
	userID	int	the user id			
Return	the store owner by the user					
insertStore()						
public int insertStore(Store store)						
Description	Insert a store.					
Parameter	Name	Type	Description			
	store	Store	the store			
Return	the new store id if success, -1 if fail					
updateStore()						
public int updateStore(Store store)						
Description	Update a store.					
Parameter	Name	Type	Description			
	store	Store	the store			
Return	the store id if success, -1 if fail					

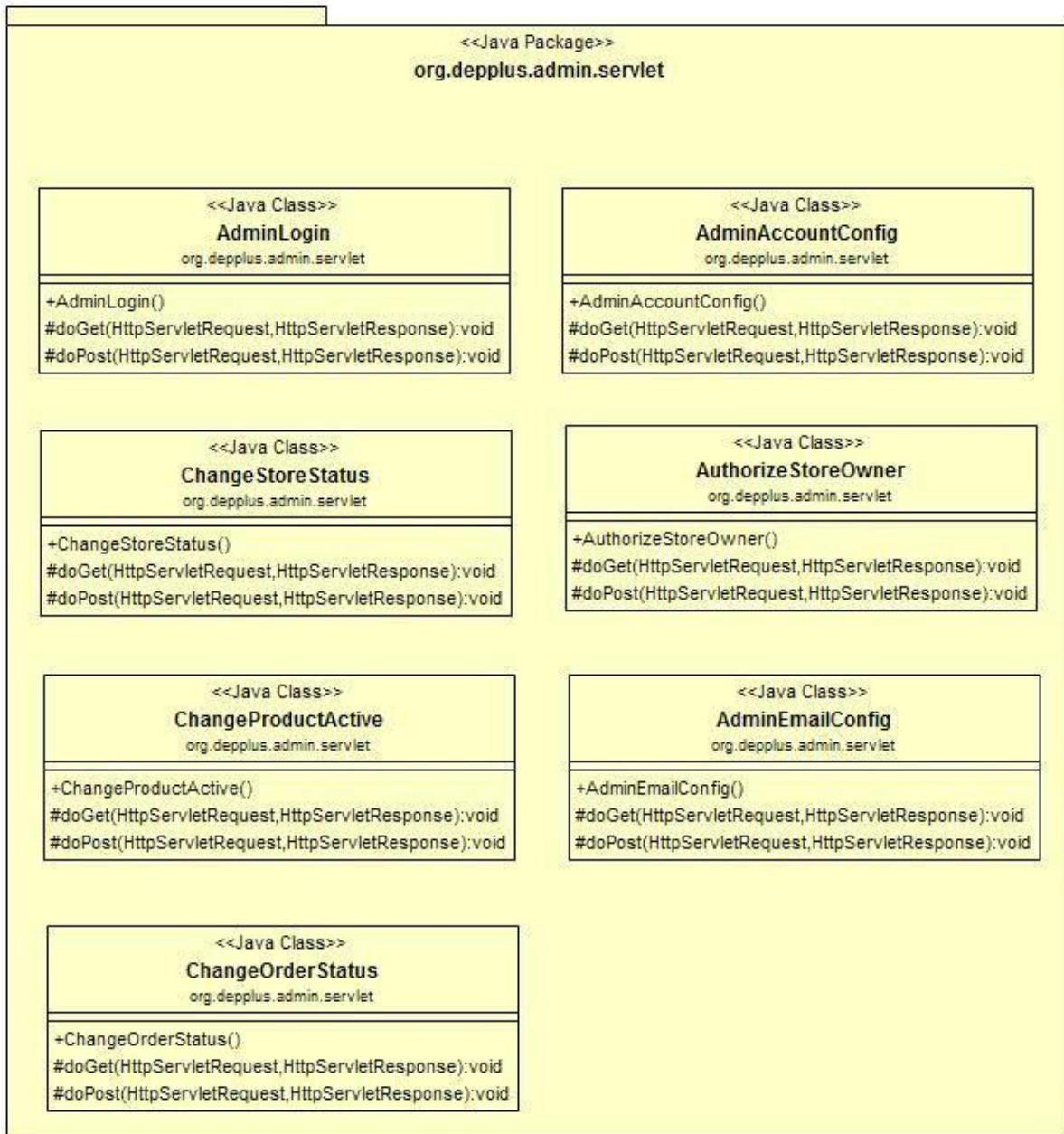
4.6.2.2.4.6 UserBusiness Class detail

Method Detail:

getUser()						
public User getUser(java.lang.String userName)						
Description	Gets the user by username.					
Parameter	Name	Type	Description			
	userName	java.lang.String	the user name			
Return	the user have the username					
getUser()						
public User getUser(int userid)						
Description	Gets the user had the corresponding id.					
Parameter	Name	Type	Description			
	userid	int	the userid			
Return	the user had the corresponding id					
Login()						
public java.lang.String login(java.lang.String username, java.lang.String password)						
Description	Login user to the system.					
Parameter	Name	Type	Description			
	username	java.lang.String	the username			
Return	password					
	java.lang.String the password "-2:-1" if username is wrong, "-1:-1" if password is wrong, "userid:isStoreOwner()" if success					
Register()						
public int register(User user)						
Description	Register an user.					
Parameter	Name	Type	Description			
	user	User	the info of the new user			
Return	the id of the new user if success, -1 if fail					

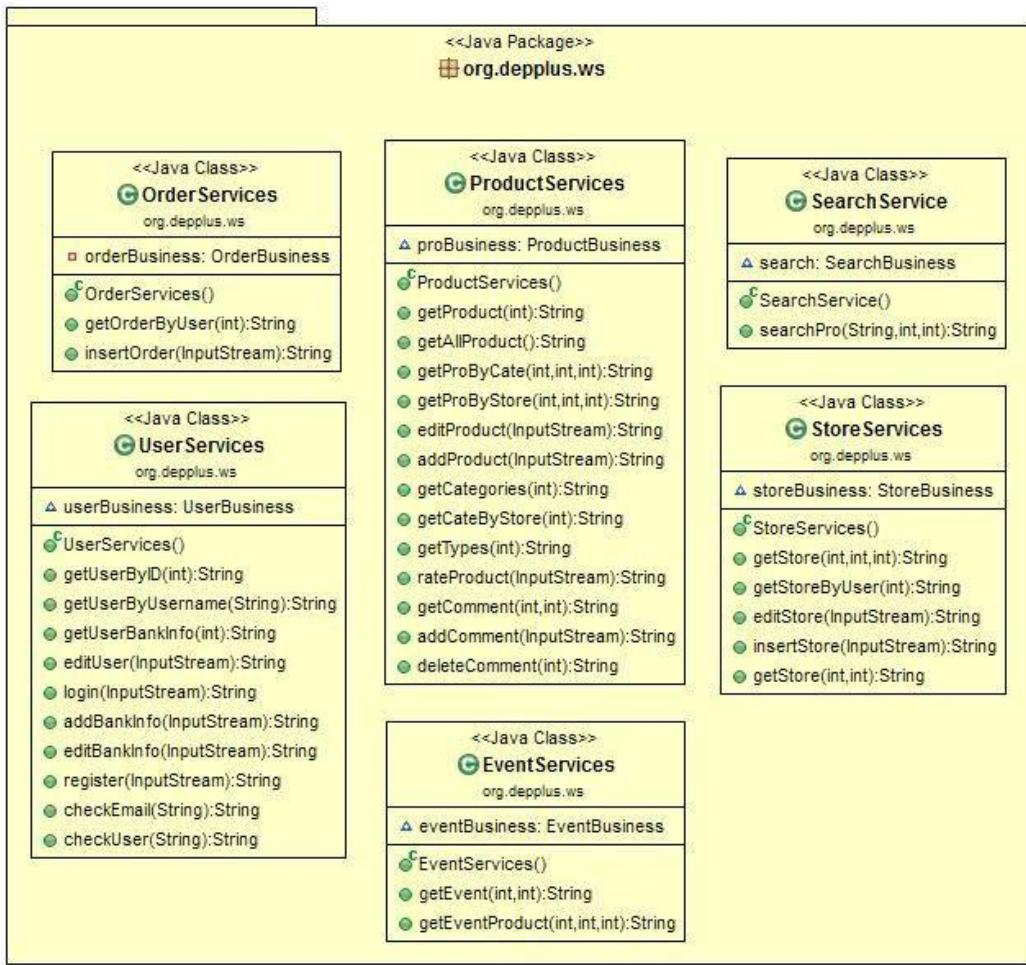
updateUser()						
public int updateUser(User user)						
Description	Update user.					
Parameter	Name	Type	Description			
	user	User	the user			
Return	the user id if success, -1 if fail.					
checkUser()						
public boolean checkUser(java.lang.String username)						
Description	Check if username existed.					
Parameter	Name	Type	Description			
	username	java.lang.String	the username to be checked			
Return	true, if successful, false otherwise.					
checkEmail()						
public boolean checkEmail(java.lang.String email)						
Description	Check if email existed					
Parameter	Name	Type	Description			
	email	java.lang.String	the email to be checked			
Return	true, if successful, false otherwise.					

4.6.2.2.5 Admin Business package:



4.6.2.2.6 Web service package:

4.6.2.2.6.1 Webservice package class diagram:



4.6.2.2.6.2 EventServices class detail:

Method Detail:

getEvent()				
URL pattern	/EventServices/GetEvent			
RESTful Type	GET			
Function	public java.lang.String getEvent(int eventID, int storeID)			
Description	Response with JSON object represent Event object when client request			
Parameter	Name	Type	Default	Description
	eventID	Int	-1	The id of the event to get

	storeID	Int	-1	The id of the store which the event belong to
Return content-type	Application/JSON			
Return description	The JSON object represent an Event object or a list of Event object as requested			
getEventProduct()				
URL pattern	/EventServices/GetEventProduct			
RESTful Type	GET			
Function	public java.lang.String getEvent(int eventID, int storeID)			
Description	Get list of all product in an event			
Parameter	Name	Type	Default	Description
	eventID	Int	-1	The id of the event
Return content-type	Application/JSON			
Return description	Return an JSON object represent a list of product in an event			

4.6.2.2.6.3 OrderServices class detail:

Method detail:

getOrderByUser				
URL pattern	/OrderServices/OrderByUser/{userID: [0-9]*}			
RESTful Type	GET			
Function	public java.lang.String getOrderByUser(int userID)			
Description	Get a list of order placed by an user			
Parameter	Name	Type	Default	Description
	userID	Int	-1	The id of the user
Return content-type	Application/JSON			
Return description	Return an JSON object represent a list of order placed by an user.			
insertOrder				
URL pattern	/OrderServices/OrderInsert			

RESTful Type	POST			
Consumes	Application/JSON			
Function	public java.lang.String insertOrder (java.io.InputStream incomingData)			
Description	Insert an order from posted data			
Parameter	Name	Type	Default	Description
	N/A	N/A	N/A	N/A
Return content-type	Application/JSON			
Return description	Return an JSON object represent the new order id if success, -1 if fail			

4.6.2.2.6.4 ProductServices class detail:

Method detail:

getProduct()				
URL pattern	/ProductServices/Products/{productID: [0-9]*}			
RESTful Type	GET			
Function	public java.lang.String getProduct(int productID)			
Description	Get a product with matched id			
Parameter	Name	Type	Default	Description
	productID	Int	-1	The id of the product
Return content-type	Application/JSON			
Return description	Return an JSON object represent the product with matched id			
getProductByCategory				
URL pattern	/ProductServices/ProByCate/{categoryID: [0-9]*}			
RESTful Type	GET			
Function	public java.lang.String getProByCate(int categoryID)			

Description	Get a list of product of a category			
Parameter	Name	Type	Default	Description
	categoryID	Int	-1	The id of the category
Return content-type	Application/JSON			
Return description	Return an JSON object represent a list of product of a category			
getProductByStore				
URL pattern	/ProductServices//ProByStore/{storeID: [0-9]*}			
RESTful Type	GET			
Function	public java.lang.String getProByCate(int storeID)			
Description	Get a list of product of a store			
Parameter	Name	Type	Default	Description
	storeID	Int	-1	The id of the store
Return content-type	Application/JSON			
Return description	Return an JSON object represent a list of product of a store			
editProduct				
URL pattern	/ProductServices/ProductEdit			
RESTful Type	POST			
Consumes	Application/JSON			
Function	public java.lang.String editProduct (java.io.InputStream incomingData)			
Description	Edit an product from posted data			
Parameter	Name	Type	Default	Description
	N/A	N/A	N/A	N/A
Return content-type	Application/JSON			
Return description	Return an JSON object represent the product id if success, -1 if fail			
addProduct				
URL pattern	/ProductServices/ProductInsert			
RESTful Type	POST			

Consumes	Application/JSON			
Function	public java.lang.String addProduct (java.io.InputStream incomingData)			
Description	Insert an product from posted data			
Parameter	Name	Type	Default	Description
	N/A	N/A	N/A	N/A
Return content-type	Application/JSON			
Return description	Return an JSON object represent the new product id if success, -1 if fail			
getCategories				
URL pattern	/ProductServices/Categories			
RESTful Type	GET			
Function	public java.lang.String getCategories(int typeID)			
Description	Get a list of all categories of a type			
Parameter	Name	Type	Default	Description
	typeID	Int	-1	The id of the type
Return content-type	Application/JSON			
Return description	Return an JSON object represent a list of all categories of a type			
getCategoryByStore				
URL pattern	/ProductServices/CateByStore			
RESTful Type	GET			
Function	public java.lang.String getcateByStore (int storeID)			
Description	Get a list of all categories of products of a store			
Parameter	Name	Type	Default	Description
	storeID	Int	-1	The id of the store
Return content-type	Application/JSON			
Return description	Return an JSON object represent a list of all categories of products of a store			
getTypeByTarget				
URL pattern	/ProductServices/Types			

RESTful Type	GET			
Function	public java.lang.String getTypes (int targetID)			
Description	Get a list of types of a target			
Parameter	Name	Type	Default	Description
	targetID	Int	-1	The id of the type
Return content-type	Application/JSON			
Return description	Return an JSON object represent			
rateProduct				
URL pattern	/ProductServices/RateProduct			
RESTful Type	POST			
Consumes	Application/JSON			
Function	public java.lang.String rateProduct(java.io.InputStream incomingData)			
Description	Rate a product			
Parameter	Name	Type	Default	Description
	N/A	N/A	N/A	N/A
Return content-type	Application/JSON			
Return description	Return an JSON object represent 1 if success, -1 if fail			
getComments				
URL pattern	/ProductServices/Comments			
RESTful Type	GET			
Function	public java.lang.String getComment(int productID)			
Description	Get a list of comments of a product			
Parameter	Name	Type	Default	Description
	productID	Int	-1	The id of the product
Return content-type	Application/JSON			
Return description	Return an JSON object represent a list of comments of a product			
addComment				

URL pattern	/ProductServices/CommentAdd			
RESTful Type	POST			
Consumes	Application/JSON			
Function	public java.lang.String addComment(java.io.InputStream incomingData)			
Description	Insert an comment from posted data			
Parameter	Name	Type	Default	Description
	N/A	N/A	N/A	N/A
Return content-type	Application/JSON			
Return description	Return an JSON object represent new comment id if success, -1 if fail			
deleteComment				
URL pattern	/ProductServices/CommentDelete			
RESTful Type	GET			
Function	public java.lang.String deleteComment(int commentID)			
Description	Delete a comment with matched id			
Parameter	Name	Type	Default	Description
	commentID	Int	-1	The id of the comment
Return content-type	Application/JSON			
Return description	Return an JSON object represent the id of the comment if success, -1 if fail			

4.6.2.2.6.5 SearchService class detail:

Method Detail

searchProduct				
URL pattern	/SearchServices/Search			
RESTful Type	GET			
Function	public java.lang.String searchPro(java.lang.String query)			
Description	Get a list of product whose name matched the query			
Parameter	Name	Type	Default	Description

	Query	String		The query
Return content-type	Application/JSON			
Return description	Return an JSON object represent a list of product whose name matched the query			

4.6.2.2.6.6 *StoreServices class detail:*

Method Detail:

getStores				
URL pattern	/StoreServices/Stores/{storeID: [0-9]*}			
RESTful Type	GET			
Function	public java.lang.String getStore(int storeID)			
Description	Get a store with matched id, get a list of all store if no id is presented			
Parameter	Name	Type	Default	Description
	storeID	Int	-1	The id of the store
Return content-type	Application/JSON			
Return description	Return an JSON object represent a store with matched id, or a list of all store if no id is presented			
getStoreByUser				
URL pattern	/StoreServices/StoreByUser/{userID: [0-9]*}			
RESTful Type	GET			
Function	public java.lang.String getStoreByUser(int userID)			
Description	Get store of an user			
Parameter	Name	Type	Default	Description
	userID	Int	-1	The id of the user
Return content-type	Application/JSON			
Return description	Return an JSON object represent the store of an user			
insertStore				
URL pattern	/StoreServices/StoreInsert			

RESTful Type	POST			
Consumes	Application/JSON			
Function	public java.lang.String insertStore(java.io.InputStream incomingData)			
Description	Insert an store from posted data			
Parameter	Name	Type	Default	Description
	N/A	N/A	N/A	N/A
Return content-type	Application/JSON			
Return description	Return an JSON object represent the new store id if success, -1 if fail			
editStore				
URL pattern	/StoreServices/StoreInsert			
RESTful Type	POST			
Consumes	Application/JSON			
Function	public java.lang.String editStore(java.io.InputStream incomingData)			
Description	Edit an store from posted data			
Parameter	Name	Type	Default	Description
	N/A	N/A	N/A	N/A
Return content-type	Application/JSON			
Return description	Return an JSON object represent the store id if success, -1 if fail			

4.6.2.2.6.7 *UserServices class detail*

Method detail:

getUserByID				
URL pattern	/UserServices/Users/{userid: [0-9]*}			
RESTful Type	GET			
Function	public java.lang.String getUserByID(int userid)			
Description	Get an user with matched id			
Parameter	Name	Type	Default	Description

	userID	Int	N/A	The id of the user
Return content-type	Application/JSON			
Return description	Return an JSON object represent an user with matched id			
getUserByUsername				
URL pattern	/UserServices/Users/{username:[a-zA-Z][a-zA-Z_0-9]*}			
RESTful Type	GET			
Function	public java.lang.String getUserByUsername(java.lang.String userName)			
Description	Get an user with matched username			
Parameter	Name	Type	Default	Description
	Username	String	N/A	The username of the user
Return content-type	Application/JSON			
Return description	Return an JSON object represent an user with matched username			
checkEmail				
URL pattern	/UserServices/CheckEmail			
RESTful Type	GET			
Function	public java.lang.String checkEmail(java.lang.String email)			
Description	Check whether an email address existed			
Parameter	Name	Type	Default	Description
	Email	String		The email address
Return content-type	Application/JSON			
Return description	Return an JSON object represent true if email address existed, false otherwise.			
checkUsername				
URL pattern	/UserServices/CheckUser			
RESTful Type	GET			
Function	public java.lang.String checkUser(java.lang.String username)			
Description	Check whether an username existed			
Parameter	Name	Type	Default	Description

	Username	String		The username
Return content-type	Application/JSON			
Return description	Return an JSON object represent true if the username existed, fail otherwise.			
editUser				
URL pattern	/UserServices/UserEdit			
RESTful Type	POST			
Consumes	Application/JSON			
Function	public java.lang.String editUser (java.io.InputStream incomingData)			
Description	Edit an user from posted data			
Parameter	Name	Type	Default	Description
	N/A	N/A	N/A	N/A
Return content-type	Application/JSON			
Return description	Return an JSON object represent the user id if success, -1 if fail			
login				
URL pattern	/UserServices/UserLogin			
RESTful Type	POST			
Consumes	Application/JSON			
Function	public java.lang.String login (java.io.InputStream incomingData)			
Description	Insert an order from posted data			
Parameter	Name	Type	Default	Description
	N/A	N/A	N/A	N/A
Return content-type	Application/JSON			
Return description	Return an JSON object represent <ul style="list-style-type: none"> • “1:1” if login successfully and user is a store owner • “1:0” if login successfully and user is a normal user • “-1:-1” if login unsuccessfully 			
registerUser				
URL pattern	/UserServices/UserRegister			

RESTful Type	POST			
Consumes	Application/JSON			
Function	public java.lang.String register(java.io.InputStream incomingData)			
Description	Insert an user from posted data			
Parameter	Name	Type	Default	Description
	N/A	N/A	N/A	N/A
Return content-type	Application/JSON			
Return description	Return an JSON object represent the new user id if success, -1 otherwise			